Effective and efficient coordination of flexibility in smart grids

Hermen A.
Toersche

Members of the graduation committee:

prof. dr. ir.	G. J. M. Smit	University of Twente (promotor)
prof. dr.	J. L. Hurink	University of Twente (promotor)
prof. dr. ir.	H. J. Broersma	University of Twente
prof. dr.	A. K. I. Remke	University of Twente & University of Münster
prof. dr. ir.	J. A. La Poutré	CWI & Delft University of Technology
prof. dr.	G. B. Huitema	TNO & University of Groningen
prof. dr. ir.	G. Deconinck	KU Leuven & EnergyVille
prof. dr.	P. M. G. Apers	University of Twente (chairman and secretary)

This research is supported by the Dutch Technology Foundation (STW), which is part of the Netherlands Organisation for Scientific Research (NWO) and partly funded by the Ministry of Economic Affairs, as part of the Dynamic real-time control of energy streams in buildings (DREAM) project (STW project number 11842).

Part of this work has been funded by EIT Digital as part of the Hybrid Energy Grid Management (HEGRID) project (EIT Digital project number 13114).

UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics and Computer Science,
Computer Architecture for Embedded Systems (CAES) and
Discrete Mathematics and Mathematical Programming (DMMP).

CTIT

Centre for Telematics and Information Technology
P.O. Box 217, 7500 AE Enschede, the Netherlands
CTIT Ph.D. thesis series no. 16–406

Copyright © 2016 Hermen A. Toersche, Enschede, the Netherlands.
Digital version (PDF) available at <http://dx.doi.org/10.3990/1.9789036541978>.
Printed by Gildeprint Drukkerijen, Enschede, the Netherlands.
Cover design by Eefke van de Haar.

ISBN 978–90–365–4197–8
ISSN 1381–3617
DOI 10.3990/1.9789036541978

EFFECTIVE AND EFFICIENT
COORDINATION OF FLEXIBILITY
IN SMART GRIDS

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. H. Brinksma,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op vrijdag 21 oktober 2016 om 14:45 uur

door

Herman Alexander Toersche
geboren op 2 april 1986
te Westerhaar–Vriezenveensewijk

Dit proefschrift is goedgekeurd door:
prof. dr. ir. G. J. M. Smit (promotor)
prof. dr. J. L. Hurink (promotor)

ABSTRACT

RENEWABLE ENERGY is starting to play a serious role in the electricity world, gradually displacing the reliable (though polluting and resource-finite) conventional electricity generation technology that has served us over the last century. However, renewables offer much less control over the production of electricity, and thereby ask for new sources of flexibility. Storage is expected to become one of the key ingredients for the further development of the energy transition, as it can bridge the gap between supply and demand in time. As a lot of renewable generation is added at the lower tiers of the grid, storage can also help to keep energy local, and thereby reduce costly grid investments and transport losses, bridging the gap between supply and demand in space.

Although physical energy storage (e.g. in batteries) is generally expensive, *demand side management* (DSM) promises to provide a different form of “storage” at (almost) no additional cost by exploiting the intrinsic flexibility within electricity consuming and producing devices. The energy transition introduces many new devices that have some flexibility in their electricity consumption or production, such as *electric vehicles* (EVs), heat pumps or *combined heat and power* (CHP) systems. What remains is to control this sea of flexibility and let the devices play their part in the *smart grid*. However, the control of devices in DSM turns out to be a hard problem, because the flexibility in devices is restricted, scattered, and there are costs associated with the use of the flexibility. To decide which devices are used (turned on or off) to reach some given goal, coordination is used to exploit the diversity of devices (in space). Furthermore, the control decisions impact the situation in the near future. To account for this, planning approaches may be used to exploit the flexibility of the devices over time. Together, this leads to a problem that is coupled in space and time, which is in general too large to be optimized directly, and should therefore be addressed in practice with heuristics or approximate methods. In this thesis, we address this DSM coordination/optimization problem.

In this context, earlier work at the University of Twente led to TRIANA as a scalable optimization and control approach for DSM in smart grids. TRIANA partitions the optimization problem according to the hierarchical structure of the electricity grid, and splits up the DSM control problem in three phases: forecasting, planning, and real-time control. Although the approach is scalable and conceptually elegant, it simplifies the problem to such an extent that the solutions are sometimes far from being optimal. Therefore, the phases of TRIANA should be

considered as dependent problems: for example, the result of real-time control depends on the forecasting and planning phases, and the planning phase should already account for this. We introduce more sophisticated planning methods (*column generation* and *profile steering*) to improve the planning results, and place these methods in a general model. To evaluate the methods, we took part in the development of an extensive simulation scenario called Flex Street. For this scenario we determine a lower bound on the cost to manage this scenario. Both of the developed planning methods bring the plan closer to the optimum than the original planning method from TRIANA (within 1 – 2% of the lower bound of the Flex Street scenario in a deterministic setting). A key strategy to keep the developed approaches scalable is a local optimization that already takes the needs of the nodes higher up in the hierarchical structure into account.

Flexible devices are in general a major source of uncertainty themselves, since their operation depends on human behaviour, which makes the forecasting of available flexibility for specific devices difficult. Dynamic dispatch approaches address this uncertainty by exploiting the interchangeability of devices, meaning that we decide just-in-time which specific devices are going to be used, e.g. with a flexibility auction. Although this dynamic dispatching makes the approach more robust against disturbances of individual devices, it also makes the reasoning about the behaviour of the system more difficult for the planning. We propose a method to plan such a system based on the simulation of the dispatch process, where the planning result determines the configuration of a controller. We evaluate the method with a subset of Flex Street, and find that the method achieves results within 2 – 10% of the lower bound, depending on the considered configuration. This approach gives robust results even with large forecast errors and a small number of devices.

To bring DSM methodologies to practice, there are still some barriers at a household level. One of these barriers is a limited standardization of the interface to flexible devices, leading to high software development and maintenance costs. A challenge in this standardization is that control methods differ in their perspective on flexibility. The *energy flexibility interface* (EFI) reacts on this challenge by proposing to communicate the structure of energy flexibility instead of a specific perspective on flexibility. We develop a comprehensive TRIANA energy application prototype that implements the EFI. The prototype supports the decentralized planning and control of real devices on low cost embedded hardware, and demonstrates that the concepts developed in this thesis are applicable in an externally given framework. It also shows that EFI maps to multiple perspectives on energy flexibility in addition to only just-in-time auction based methods.

Concluding, this work lays a foundation for the further development of a flexible, effective and efficient coordination approach for flexibility in smart grids, bringing the dream of DSM – and thereby the cost effective implementation of the energy transition – a bit closer to reality.

SAMENVATTING

CONVENTIONELE ELEKTRICITEITSCENTRALES hebben de laatste honderd jaar gezorgd voor een betrouwbare maar ook vervuilende en uitputbare energievoorziening. Tegenwoordig worden deze centrales steeds meer verdrongen door duurzame opwekkers. Deze opwekkers brengen echter onzekerheid en veel minder stuurbaarheid met zich mee, wat vraagt om nieuwe bronnen van flexibiliteit. In de toekomst van de energietransitie gaat opslag een hoofdrol spelen om tekorten aan vraag en aanbod over tijd te overbruggen. Daarnaast kan opslag energie dicht bij de plek van opwek houden, en daarmee vraag en aanbod in ruimte dicht bij elkaar brengen. Dit bespaart investeringen en verliezen in het elektriciteitsnet.

De opslag van elektriciteit, bijvoorbeeld in accu's, is vooralsnog erg duur. *Vraagsturing* biedt een goedkoop alternatief voor opslag door de flexibiliteit binnen apparaten zelf te benutten. Veel nieuwe apparaten, zoals elektrische auto's, HRe-ketels en warmtepompen, beschikken over zulke flexibiliteit. In principe hoeven we deze apparaten alleen nog maar goed aan te sturen, en ze als onderdeel van een *smart grid* (intelligent elektriciteitsnet) te beschouwen. Deze aansturing blijkt makkelijker gezegd dan gedaan, omdat de flexibiliteit versnipperd, beperkt en niet helemaal gratis is. Op elk moment moet afgestemd worden welke apparaten aan- en uitgezet worden, gebruikmakend van de verschillen tussen apparaten (diversiteit in ruimte). Omdat deze beslissingen ook de nabije toekomst beïnvloeden, is het raadzaam om tevens het verbruik over tijd te beschouwen. Door te plannen kan hier rekening mee gehouden worden, en kan ook diversiteit over tijd benut worden. Samen leidt dit echter tot een aansturingsprobleem dat gekoppeld is in tijd en ruimte, wat meestal te complex is om als geheel te optimaliseren. Daarom wordt dit probleem vaak aangepakt met heuristieken en schattingen. Dit proefschrift gaat over zulke optimalisatiemethodes voor vraagsturing.

In eerder werk aan de Universiteit Twente is TRIANA ontwikkeld, een schaalbare aanpak voor de aansturing en optimalisatie van smart grids met vraagsturing. TRIANA deelt het aansturingsprobleem op volgens de hiërarchische structuur van het elektriciteitsnet, en splitst het vraagsturingsprobleem in drie fases: voorspelling, planning en real-time aansturing. Hoewel de eerder ontwikkelde aanpak inderdaad schaalbaar en conceptueel elegant is, blijkt deze het probleem dusdanig te vereenvoudigen dat de aansturing soms verre van optimaal is. Om dit te verhelpen zouden de fases van TRIANA als gekoppelde problemen beschouwd moeten worden: de mogelijkheden van de aansturing hangen af van de voorspelling en de

planning, en vice versa zou de planner ook al rekening moeten houden met het gedrag van de aansturing. In dit proefschrift verbeteren we de planning door twee vooruitstrevende planningsmethodes aan TRIANA toe te voegen (*kolomgeneratie* en *profielsturing*), en plaatsen we deze methodes samen met de oude methode in een gemeenschappelijk raamwerk. Om deze methodes goed te kunnen beoordelen hebben we deelgenomen aan de ontwikkeling van Flex Street, een groot simulatiemodel voor vraagsturing in smart grids. Voor dit scenario bepalen we een ondergrens voor de best mogelijke aansturing, en we laten zien dat de ontwikkelde planningsmethodes dicht bij het optimum komen (zonder onzekerheid meestal binnen 1 – 2% van de ondergrens). De schaalbaarheid van de methodes wordt verbeterd door lokaal te anticiperen op de wensen hoger in de hiërarchie.

De flexibiliteit van apparaten is afhankelijk van de manier hoe mensen ze gebruiken, en vormt daarmee een bron van onzekerheid op zich. Het is vaak lastig te voorspellen of een specifiek apparaat op een gegeven moment beschikbaar zal zijn. Om dit probleem te omzeilen gebruikt men vaak aansturingsmethodes die pas op het laatste moment besluiten welke apparaten ingeschakeld dienen te worden, bijvoorbeeld volgens een veilingprincipe. Hoewel deze aanpak robuuster is tegen individuele voorspellingsfouten, zorgt deze er ook voor dat het planningsprobleem een stuk lastiger wordt. Wij lossen dit probleem op door te simuleren hoe een groep apparaten reageert op de aansturing met verschillende instellingen over tijd, waarmee we een soort van planning voor zo'n systeem bepalen. Experimenten met (een klein deel van) Flex Street laten zien dat de aanpak in veel gevallen binnen 2 – 10% van de theoretische ondergrens presteert, zelfs in gevallen met kleine groepen en grote voorspellingsfouten.

Om vraagsturing op huishoudniveau in de praktijk te brengen zullen nog enige hindernissen overwonnen moeten worden. Een van die hindernissen is een beperkte standaardisatie van de ontsluiting van apparaatflexibiliteit, wat leidt tot hoge softwareontwikkelkosten en -onderhoudskosten. Een van de uitdagingen bij deze standaardisatie is dat aansturingsmethodes verschillende perspectieven op flexibiliteit hanteren. De *energieflexibiliteitsinterface* (EFI) speelt hier op in door de structuur van de flexibiliteit te ontsluiten, in plaats van het perspectief voor een specifieke aansturingsmethode. Wij hebben een vrij compleet prototype van een EFI-ondersteunende TRIANA-energietoepassing ontwikkeld, waarmee we in staat zijn om op goedkope embedded hardware echte apparaten decentraal aan te sturen. Dit prototype toont aan dat onze concepten toepasbaar zijn in een extern bepaalde omgeving, en laat tevens zien dat de EFI inderdaad verenigbaar is met een ander dan veilinggebaseerd perspectief op flexibiliteit.

Tot slot concluderen we dat dit proefschrift een basis legt voor de verdere ontwikkeling van een flexibele, effectieve en efficiënte aanpak voor de afstemming van flexibiliteit in smart grids, waarmee we de verwachtingen van vraagsturing – en daarmee de betaalbare voltooiing van de energietransitie – een stukje dichterbij de werkelijkheid brengen.

DANKWOORD

EEN PROMOVEDUS wordt veroordeeld tot het schrijven van een proefschrift, waarbij strafvermeerdering eerder regel dan uitzondering is. Het samenstellen van dit boekje heeft dan ook flink wat frustratie en jaren van mijn leven gekost. Met een vrijlating in het vooruitzicht is dit een goed moment om eens terug te blikken op deze tijd, en de bezoekers die het iets dragelijker hebben gemaakt te bedanken.

Hoewel het gebruikelijk is om hiermee tot het laatst te wachten, begin ik voor de verandering eens dicht bij huis met het bedanken van mijn familie. In het bijzonder bedank ik mijn ouders, broers Robert en Ruben, en aanhang Annekarlijn en Eefke (ook dank voor het compleet renoveren van de omslag!) voor hun verwoede pogingen om me de afgelopen jaren – al dan niet met succes – van het werk te houden, waarbij middelen zoals gezelligheid, tosti's en vakanties naar Zuid-Europa niet geschuwd worden. Ondanks dat ze me voor gek verklaren dat ik ooit ben gaan promoveren, zorgen ze toch voor de omgeving waarin dit mogelijk is.

Iets verder van huis vinden we de “Wierden”-vrienden van Het Noordik, met wie ik nog geregeld verstraagde kerstdiners, levensgevaarlijke barbecues en cultureel verantwoorde vakanties mag delen. Van hen hebben Johan en Wouter de taak aanvaard om mij bij te staan tijdens mijn verdediging, en indien nodig de verdediging over te nemen (dus jongens, begin maar vast met lezen). Wouter noem ik bij deze nog een keer omdat hij gevraagd had of hij in mijn boekje genoemd kon worden. We gaan er de 21e een feestje van bouwen, en ik hoop dat we onze tradities nu “het echte leven” toch echt begint nog een flinke tijd voort kunnen zetten.

Tijdens het promoveren zie je je collega's eigenlijk nog het meest. De sfeer en can-do-mentaliteit van CAES met sterren zoals dr. Phil zijn dan ook een belangrijke motivatie geweest om na mijn afstudeertraject te blijven plakken. De rondjes volgens vast stramien (vaak met discussies over de projectjes van en met Jordy) en de escalerende koffiepauzes dragen hier erg aan bij, hoewel ik deze de laatste jaren helaas iets te vaak gemist heb. Ook staat er altijd wel een lotgenoot of ervaringsdeskundige paraat voor een sessie Oudhollandsche spelen met kantoorartikelen als je weer eens geen zin hebt om te schrijven (o.a. Westeros, Blomski en the Korevator).

De aanstichter, aanvoerder en eveneens deelnemer aan dit illustere gezelschap is Gerard, wie altijd goede slachtoffers voor de groep weet te charteren (mijzelf inclusief) en niet bang is om ook praktische projectjes uit te delen, variërend van het hacken van slimme meters voor een startup tot het marketen van de software van de concurrent in de VS. Bedankt voor de gave groep en de onbegrensde mogelijkheden. Over tijd is het zwaartepunt van mijn onderzoek meer richting Johann

geschoven, wie als geen ander systeemvisie heeft, écht opbouwende (maar helaas onleesbare) kritiek levert en precies aanvoelt wat er nodig is om mensen zichzelf te laten overtuigen. Zonder de inbreng van Johann had het proefschrift er heel anders uitgezien. Johann, bedankt!

Marlous en Nicole (en eerder ook Thelma) zijn essentieel voor het debuggen van vastlopende UT-processen. Tevens bedank ik jullie voor het regelen van ~~xxxxxx~~ conferentiebezoeken, fancy vakgroepuitjes en wat er allemaal nog meer komt kijken bij het universiteitsleven. Vanuit DMMP – waar ik eigenlijk vooral kom als er taart of feest is – heeft Marjo die rol vervuld, waarvoor dank. Daarnaast dank ik Bert voor het temmen van onze ICT-middelen, welke we soms flink mishandeld hebben voor allerlei doeleinden.

De aanvang van mijn promotie betekende een overstap naar de energiesekte binnen CAES/DMMP, die destijds alleen nog uit Albert, Vincent en Maurice (en Stefan?) bestond. Zij mogen als voorvaders van TRIANA hier natuurlijk niet ontbreken. De twee-eenheid Albert en Vincent heeft in een moordtempo een plek voor Twente in de energiewereld opgeëist, en heeft mede gezorgd voor het leger promovendi dat we nu hebben in het energiehok. Maurice de wiskundige heeft met zijn werk de basis gelegd voor Hoofdstuk 3; zijn muziek“smaak” zal mij nog lang heugen. Stefan heeft ons geregeld getraakteerd op uitstapjes naar allerlei gave projecten in Duitsland, en natuurlijk de Bierbörse. Marco was de afgelopen drie jaar mijn eerste aanspreekpunt voor rare ideeën, en verdient daarom veel meer ruimte dan ik hem hier kan geven. Ook energiekelingen Gerwin, Thijs, Richard, Jiří, le Jonathan, Nikola, Bart, Diego, Krystian, Maryam, Gijs, Marijn, James en een hele rits afgestudeerden hebben in verschillende mate bijgedragen aan dit boekje.

Het merendeel van mijn promotie is betaald door Technologiestichting STW, die ons alle ruimte en vertrouwen heeft gegeven om nuttige dingen te doen. Eigenlijk besef je pas achteraf wat voor luxepositie dit is. Bedankt en houd dit vol!

Een belangrijk deel van dit boekje is “veroorzaakt” door Felix, wie vijf jaar geleden als afgestudeerder uit Utrecht een vergelijking organiseerde tussen smart grid-aansturingsmethoden, wat een mooi artikel heeft opgeleverd (zie Appendix B), evenals verschillende memorabele meetings en een basis voor het afgestudeerwerk van Marilena (zie Appendix B.5). We gebruiken deze Felixcase tegenwoordig nog steeds vaak, wat lang niet alle afgestudeerden kunnen zeggen. Bedankt!

Tijdens Felix hebben we Bert ontmoet, een geniale eigenzinnige onderzoeker pur sang met wie ik het genoeg heb gehad een maand samen te mogen werken bij VITO en EnergyVille in België. Bert heeft me samen met zijn collegiale collega's van de energie-afdeling en de club uit Leuven een mooie en ook erg leerzame tijd bezorgd, die de basis gelegd heeft voor Hoofdstuk 4. Bedankt Bert!

Stergios heeft met zijn stage en afstuderen bij TNO een aanzet gegeven voor Hoofdstuk 5, welke in het HEGRID-project verder is uitontwikkeld. Naast Stergios bedank ik in het bijzonder Mente voor de prettige samenwerking.

Daarom, kortom: iedereen bedankt en veel succes met lezen!

Hermen

Westerhaar–Vriezenveensewijk, september 2016

CONTENTS

1	Introduction	1
	<i>problem statement 2 · contributions 3 · outline 3</i>	
2	Coordination in smart grids (related work)	5
	<i>background 6 · coordination 8 · TRIANA 30</i>	
3	Coordination in time	41
	<i>model 45 · IDDP 66 · column generation 74 · profile steering 107 · experiments 117</i>	
4	Coordination in space	133
	<i>supply model 148 · metaheuristics 153 · use case 158 · experiments 172</i>	
5	Coordination in practice	191
	<i>EFI/EF-PI 193 · TRIANA–EF-PI 198 · control space adapters 203 · experiments 219</i>	
6	Conclusion	229
	<i>summary 229 · conclusions 232 · contributions 234 · recommendations 235</i>	
A	The world of energy (background)	239
	<i>energy transition 239 · smart energy 250</i>	
B	Flex Street	273
	<i>description 275 · model 282 · simulations 288 · case study 293 · aggregate model 296</i>	
C	Implementation details	307
	<i>coordination in time 307 · space 312 · practice 316</i>	
	References	319
	Publications	345
	Glossary	347
	Symbols	353

1	INTRODUCTION	1
1.1	Problem statement	2
1.2	Contributions	3
1.3	Outline	3
2	COORDINATION IN SMART GRIDS (RELATED WORK)	5
2.1	Introduction	5
2.2	Background	6
2.2.1	Conventional power systems	6
2.2.2	Energy transition	6
2.2.3	Smart energy systems	7
2.3	Coordination	8
2.3.1	Conventional coordination	10
2.3.2	Smart grid coordination	18
2.4	TRIANA	30
2.4.1	Planning	32
2.4.2	Operational control	35
2.4.3	Evaluation	38
2.5	Conclusion	39
3	COORDINATION IN TIME	41
3.1	Introduction	41
3.2	Hierarchical model	45
3.2.1	Introduction	45
3.2.2	Overview	47
3.2.3	Node model	51
3.2.4	Multicommodity/hybrid energy management	57
3.2.5	Scalability	62
3.3	Price steering with IDDP	66
3.3.1	Introduction	66
3.3.2	Problem statement	67
3.3.3	Algorithm	68
3.3.4	Price updates	68
3.3.5	Experiments	72
3.3.6	Evaluation	73
3.4	Column generation	74
3.4.1	Introduction	74
3.4.2	Dantzig–Wolfe decomposition and column generation	76
3.4.3	Algorithm	81
3.4.4	Experiments	93
3.4.5	Evaluation	104
3.5	Profile steering	107
3.5.1	Introduction	107
3.5.2	Problem statement	109
3.5.3	Algorithm	111
3.5.4	Experiments	114
3.5.5	Evaluation	116

3.6	Year simulation experiments	117
3.6.1	Introduction	117
3.6.2	Configuration	118
3.6.3	Results	121
3.6.4	Evaluation	127
3.7	Conclusion	129
3.7.1	Recommendations	131
4	COORDINATION IN SPACE	133
4.1	Introduction	133
4.2	Problem statement	135
4.2.1	Informal description	135
4.2.2	Formal model	141
4.2.3	Structured supply curve offer model	148
4.2.4	Overview	151
4.3	Optimization with metaheuristics	153
4.3.1	Overview	154
4.3.2	Random optimization	155
4.3.3	Simulated annealing	156
4.3.4	Genetic algorithm	157
4.4	Example use case with heat pumps	158
4.4.1	Device model	159
4.4.2	Binary (on/off) devices	160
4.4.3	Objective	160
4.4.4	Deterministic variant	162
4.4.5	Stochastic variant	163
4.4.6	Heat pump bidding function	164
4.5	Experiments	172
4.5.1	Search parameters	172
4.5.2	Use case results	173
4.5.3	Evaluation	183
4.6	Conclusion	185
4.6.1	Recommendations	186
5	COORDINATION IN PRACTICE	191
5.1	Introduction	191
5.2	EF-Pi: a flexible energy management platform	193
5.2.1	The interoperability challenge	193
5.2.2	Energy flexibility interface and platform	195
5.3	TRIANA–EF-Pi	198
5.3.1	Introduction	198
5.4	TRIANA–EF-Pi control space adapters	203
5.4.1	Uncontrolled CSA	204
5.4.2	Time Shiftable CSA	204
5.4.3	Buffer CSA	209
5.4.4	Unconstrained CSA	218

5.5	Experiments	219
5.5.1	Scheduling performance	219
5.5.2	Planning resource use	220
5.5.3	Multicommodity optimization case	222
5.6	Conclusion	226
5.6.1	Recommendations	227
6	CONCLUSION	229
6.1	Summary	229
6.1.1	Coordination in smart grids	229
6.1.2	Coordination in time	230
6.1.3	Coordination in space	230
6.1.4	Coordination in practice	231
6.2	Conclusions	232
6.3	Contributions	234
6.4	Recommendations	235
6.4.1	Better optimization methods	235
6.4.2	Different time scales	235
6.4.3	Security and dependability	236
6.4.4	Guaranteed performance and capacity planning	237
6.4.5	Forecast what we can	237
6.4.6	Cost reduction	238
A	THE WORLD OF ENERGY (BACKGROUND)	239
A.1	Energy transition	239
A.1.1	Renewables	240
A.1.2	Electrification	244
A.1.3	Decentralization	248
A.1.4	Storage	249
A.2	Smart energy systems	250
A.2.1	Hybrid energy systems	251
A.2.2	Smart grids	252
B	FLEX STREET	273
B.1	Introduction	273
B.2	Case description	275
B.2.1	Controllable devices	277
B.2.2	Uncontrollable devices	278
B.2.3	Hot tap water model	278
B.2.4	Uncertainty	278
B.2.5	Objective	280
B.3	Mathematical model	282
B.3.1	Introduction	282
B.3.2	Device models	282
B.4	Results	288
B.4.1	Discussion	291
B.4.2	Conclusion	292

B.5	A case study based on Flex Street	293
B.5.1	Introduction	293
B.5.2	Scenario	293
B.5.3	Results	295
B.6	Aggregate lower bound model	296
B.6.1	Aggregate model	296
B.6.2	Device models	297
B.6.3	Results	300
B.7	Conclusion	306
B.7.1	Recommendations	306
C	IMPLEMENTATION DETAILS	307
C.1	Coordination in time	307
C.1.1	Exploiting multiple commodity providers (quadratic case)	307
C.1.2	Local search for pattern combinations	309
C.2	Coordination in space	312
C.2.1	Parameters of metaheuristics experiments	312
C.3	Coordination in practice	316
C.3.1	TRIANA–EF-Pi protocol for decentralized control	316
	REFERENCES	319
	PUBLICATIONS	345
	GLOSSARY	347
	SYMBOLS	353

FIGURES

<i>1. Introduction</i>	
1.1	Overview of this thesis 4
<i>2. Coordination in smart grids (related work)</i>	
2.1	Constraint aggregation 23
2.2	SDM double-sided Walrasian auction 25
2.3	Partitioned planning approach in TRIANA 31
2.4	Realization of planning hierarchy in TRIANA 33
2.5	House cost control energy stream model 35
<i>3. Coordination in time (TRIANA)</i>	
3.1	Basic overview of decoupled coordination 43
3.2	Partitioned planning approach in TRIANA (repeats Figure 2.3) 50
3.3	Realization of planning hierarchy in TRIANA 51
3.4	General node in hierarchical node model 52
3.5	Numerical example of hierarchical model ($\tau = 1$ h, $n_t = 5$) 56
3.6	Exponential growth of iteration count in tree height 63
<i>3.3. IDDP</i>	
3.7	Node configuration of IDDP in experiments 72
3.8	Planned demand in planning session at 12:00 with different methods 73
<i>3.4. Column generation</i>	
3.9	Dantzig–Wolfe block angular structure in linear programs 78
3.10	Node configuration of column generation in experiments 96
3.11	Nested node configuration with pass-through subproblems 98
3.12	Convergence of column generation with subgroups 99
3.13	Nested configuration with pass-through subproblems 100
3.14	Nested configuration with multiple integer rounds 100
3.15	Integrality penalty by $ \mathcal{I}_{\mathcal{N}} $, with MIP, $\mathbf{n}_{k,\mathcal{N},\mathbf{r}} = \langle 10 \rangle$, and $n_{k,\mathcal{N},\mathbf{I}} = 1$ 101
3.16	Integrality penalty by column selection method and child count 101
3.17	Nested node configuration for experiment of Section 3.4.4.5 103
3.18	Nested column generation top- and bottom-level convergence 104
<i>3.5. Profile steering</i>	
3.19	Requested difference profiles and responses 108
3.20	Node configuration of profile steering in experiments 115
3.21	Planned demand during profile steering planning 115

3.6. Year simulation of Flex Street	
3.22 Load duration curves with different planning methods	123
3.23 Difference duration curves with different planning methods	124
3.24 Demand profile around critical peak period	124
4. Coordination in space (auction dispatch)	
4.1 Demand curve aggregation	138
4.2 Supply curve and clearing	138
4.3 Example supply curves	140
4.4 Supply curve correction when curves never meet	144
4.5 Simplified supply curve representation	150
4.6 Overview of dispatch centric optimization	151
4.7 Metaheuristic optimization overview	155
4.8 SoC-based bidding curve with static flexible range	165
4.9 SoC-based bidding curve with dynamic flexible range	165
4.10 l and n/m device bidding strategy	168
4.11 Exchange profile (deterministic)	174
4.12 Metaheuristic search convergence (deterministic)	174
4.13 Exchange profile (stochastic)	179
4.14 Metaheuristic search convergence (stochastic)	179
4.15 Exchange profile (stochastic) with different prices	182
5. Coordination in practice (TRIANA-EF-Pi)	
5.1 $n \times m$ compiler problem	194
5.2 EFPi user interface screenshot	199
5.3 Connection manager screenshot	199
5.4 Household deployment	201
5.5 Time Shiftable discretization	205
5.6 Time Shiftable DP	205
5.7 Buffer automata product	211
5.8 Buffer fill level function merging	211
5.9 Buffer CSA τ projection	213
5.10 Buffer scheduling performance	221
5.11 Benchmark Buffer product automaton	221
5.12 Buffer computational performance	221
5.13 Hybrid heat pump configuration	223
A. The world of energy (background)	
A.1 Retail, feed-in tariffs over the years	240
A.2 House profile with high electrification	245
A.3 Residential storage, greedy versus grid-aware control	266
B. Flex Street	
B.1 Flex Street model of house with all controllable devices	276
B.2 Heat demand in Flex Street	279
B.3 Load duration curves of TRIANA and Intelligator	289
B.4 Model of case study	294

B.5	Maximum-so-far in case study (2013, CPP)	295
B.6	Aggregate energy flexibility of uncontrollable load in Flex Street	301
B.7	Aggregate energy flexibility of time shiftables in Flex Street	301
B.8	Aggregate energy flexibility of PHEVs in Flex Street	302
B.9	Aggregate energy flexibility of batteries in Flex Street	302
B.10	Aggregate energy flexibility of heating in Flex Street	303
B.11	Comparison of peak minimization and quadratic solutions	303
	<i>C. Implementation details</i>	
C.1	Example iteration of local pattern combination search	310

TABLES

<i>3. Coordination in time (TRIANA)</i>	
3.1	Year simulation results with Flex Street (Moderate variant) 125
<i>4. Coordination in space (auction dispatch)</i>	
4.1	Results for deterministic variant 176
4.2	Results for stochastic variant 181
4.3	Results for stochastic variant (swapped scenarios) 183
<i>5. Coordination in practice (TRIANA-EF-Pi)</i>	
5.1	Simulation results of multicommodity optimization case 225
<i>B. Flex Street</i>	
B.1	Penetration of controllable devices in Flex Street households 275
B.2	Parameters of controllable devices in Flex Street 275
B.3	Line loss transport costs in simplified LV and MV model 289
B.4	Performance of different control approaches 290
B.5	Parameters of scenarios in Appendix B.5 294

INTRODUCTION

“[In the US,] the estimated net investment needed to realize the envisioned power delivery system (PDS) of the future is between \$338 and \$476 billion [over a period of 20 years]. The total value estimate range of between \$1294 and \$2028 billion.”

Estimating the costs and benefits of the smart grid, EPRI, 2011 [126]

“...demand response potentially is one of the most promising low cost instruments that provides an alternative source of flexibility and brings substantial benefits in the integration of [renewables].”

Integration of renewable energy in Europe, DNV GL, 2014 [80]

problem statement 2 · contributions 3 · outline 3

FLEXIBILITY IS IMPORTANT to keep the electricity system reliable and affordable, and has conventionally been offered mostly by controlling power plants. However, renewable energy sources are starting to replace these power plants in daily operation, yet do not offer the flexibility that we need to have. While it is in principle possible to keep these power plants available for the times that renewables can not satisfy the demand, this is very costly, and gives a barrier to a complete transition to renewables. Therefore, we need new sources of flexibility in the electricity system.

Demand side management (DSM), or *demand response* (DR), provides a promising alternative source of flexibility. The idea of DSM is that we can shift the electricity demand of devices over time, making use of the flexibility that is already available (or can be easily added) within devices. Especially the new large residential energy consuming devices, such as heat pumps and *electric vehicles* (EVs), have a lot of intrinsic storage capacity. If we can exploit this capacity, then we have a very large and virtually free storage resource. DSM is widely considered to be a key ingredient for a *smart grid*, which promises an efficient, reliable and affordable energy system.

Although DSM seems ideal at first, there are some caveats. A major challenge with this DSM resource is that it is split up in very small pieces and dispersed over a large area. This means that we have to decide “who does what” to make the devices act in unison, which is easier said than done. Devices differ vastly in their characteristics, and are used in different ways. The dispersal of devices leads

to locality: for example, a household may locally consume the production of its own solar panels to save transport costs, but should export it when the demand is more needed elsewhere. Also, some devices prefer to be served as soon as possible (e.g. to make an EV available for use), whereas others prefer to be served as late as possible (e.g. to reduce leakage). At the same time, we have to make sure that our decisions do not lead to problems in the future, as we may not postpone demand indefinitely. Together, this leads to a complex “who does what and when” energy planning and coordination puzzle at an unprecedented scale.

This thesis addresses the optimization of DSM flexibility over time and between devices (in space). As there are so many devices, we have to solve the problem in a scalable way. Furthermore, the efficiency is important to keep the system affordable: the quoted reference [80: p. xvii] notes that “the net benefits of demand response will remain positive at least for those types of DR that can be used and activated at limited cost”. This means that we also have to pay attention to the more practical aspects of DSM, and e.g. can not afford to install and maintain an industrial grade controller for every device. It also means that we have to account for the costs to use the flexibility of devices, e.g. due to higher wear or intangible costs such as discomfort. We consider approximate solutions methods, and see optimization inaccuracies as economical losses.

A major theme in this new energy world is uncertainty, which makes the planning of devices even more difficult. Although the forecasting of renewable energy production is improving, still significant uncertainty remains about the time, volume and location of their feed-in. Less acknowledged is that the availability of devices is uncertain as well, and the forecasting of their flexibility is largely unexplored. Hybrid energy systems can help to cope with the associated risks, and may also relieve the electricity system at peak times more generally. Therefore, we develop optimization methods that can exploit hybrid devices.

Before we go into the content matter, we first present a structured description of the problems that we address in this thesis, our contributions towards solving these problems, and the structure of the rest of this thesis.

1.1. PROBLEM STATEMENT

This thesis addresses the following central problem:

How can we effectively and efficiently coordinate the flexibility in smart grids in time and space?

To tackle this central problem, we split it in the following subproblems:

- How can we effectively and efficiently optimize the behaviour of large groups of devices in time and space?
- How does uncertainty affect efficient and effective coordination, and how can we cope with this uncertainty?
- Can we use these optimization methods in a practical context?

1.2. CONTRIBUTIONS

The contributions of this thesis are as follows:

- A composable and extensible *framework* for the hierarchical, temporally coupled optimization of DSM; Chapter 3
 - Efficient and effective optimization *algorithms* based on this framework;
 - Support for *multicommodity* energy streams in the optimization;
 - A comparative *evaluation* of these algorithms against each other and a lower bound solution;
- An optimization *algorithm* for a Walrasian auction based coordination method based on the simulation of the dispatch process that is robust under uncertainty; Chapter 4
- A proof-of-concept DSM system *implementation* using the algorithms of Chapter 3 on a practical management platform for energy flexibility. Ch. 5

1.3. OUTLINE

This thesis proceeds as follows (the numbers correspond to chapter numbers):

2. Coordination in smart grids discusses related work on coordination methods for the conventional grid and smart grids. We discuss and review TRIANA, the coordination approach that we develop further in Chapter 3. p. 5
3. Coordination in time follows a more conventional approach for the hierarchical decentralized optimization of large groups of devices. We set up a common model for these optimization algorithms, and present three algorithms with different characteristics. We compare these algorithms to each other and to a lower bound solution in a large scale simulation case. p. 41
4. Coordination in space considers more dispatch oriented coordination methods, which are popular in the context of smart grids. We propose a simulation based method to optimize the demand of a group of devices that is dispatched with a Walrasian auction based coordination method, and compare it to a lower bound solution in a small scale simulation case. p. 133
5. Coordination in practice presents an implementation of TRIANA on the EF-Pi energy management platform, which demonstrates the techniques from Chapter 3 in a realistic environment. p. 191
6. Conclusion concludes the thesis, addresses the research questions in light of the contributions, and presents recommendations for future work. p. 229

Consult the appendices for further reading and for reference:

- A. *The world of energy* gives a background on the basics of smart grids, and is recommended for readers with a limited energy background. p. 239
- B. *Flex Street* describes a large simulation case that we use throughout this thesis to demonstrate the algorithms that we have developed. p. 273
- C. *Implementation details* provides details about various topics that are of interest mainly for those who want to reproduce the results in this thesis. p. 307

The chapters are related by the structure that we present in Figure 1.1.

FIGURE 1.1: Overview of this thesis.

COORDINATION IN SMART GRIDS

2

Abstract

Changes in the energy domain make the control and coordination of energy resources increasingly important. We discuss these energy domain changes, and review smart energy systems which accommodate these changes. In many of these solutions, the coordination of energy streams is essential, which leads to the smart grid concept. We review coordination approaches that use this concept, and extend one of these approaches, TRIANA, in the remainder of this thesis. We observe that the TRIANA approach is not robust against uncertainty, and propose directions for improvement. In the studied approaches – including TRIANA, predictability is a key issue that needs to be addressed.

background 6 · coordination 8 · TRIANA 30

2.1. INTRODUCTION

THE BEGINNING OF THIS CENTURY (2000 – 2016) gave a glimpse of the energy transition, which is expected to further unfold in the coming decades. This transition is characterized by trends of increasing renewable generation, further electrification, energy storage, and decentralization. These trends are motivated by a combination of environmental concerns, energy economics, and politics. To support these trends, large changes to the energy infrastructure that require large investments are inevitable.

Over the last decades, the cost and capabilities of microelectronics and power electronics have improved dramatically. These improvements shift the balance from the use of “real” energy resources – thicker cables, larger generators, more fuel – towards putting more effort in improving the use of these resources. Many of the innovative energy systems that follow this shift are described as *smart energy systems* (SES). The *smart grid* paradigm applies this concept to the electricity infrastructure, but can also be applied in e.g. gas or heat infrastructures. By making better use of energy resources, smart grid technology can reduce the need for investments in energy transportation hardware. For example, a controller that reduces the peak load on a transformer allows the use of a smaller transformer, or extends the lifetime of an existing one.

The coordination of energy streams plays a crucial role in many smart grid technologies. Over the last years, smart grid coordination problems have attracted a lot of interest from research and industry, which has resulted in various control approaches. Furthermore, we observe that similar problems have been addressed over the last decades in conventional power systems optimization.

We discuss the background on smart grid control as follows. First, we defer a discussion on the context in which smart grids operate to Appendix A, as many readers will already be familiar with most of the context. This appendix discusses the energy transition (Appendix A.1), smart energy systems (Appendix A.2), and smart grids (Appendix A.2.2). We provide a short summary of this context in Section 2.2. Smart grids give interesting coordination problems: in Section 2.3, we review these coordination problems, and solutions that have been proposed in literature. Finally, we dedicate Section 2.4 to TRIANA, the specific smart grid coordination approach that will be the focus of the following chapters of this thesis. We compare it to literature, observe that the current configuration is not well suited for environments with uncertain resources, and identify directions for improvement. In the rest of the thesis, we explore some of these improvements.

2.2. THE SMART ENERGY CONTEXT

In Appendix A, we give an overview of the context in which smart grids operate. Here, we provide a short summary.

2.2.1. CONVENTIONAL POWER SYSTEMS

Conventional power systems follow a hierarchical structure, in which power flows from a set of large power plants, through the transmission and distribution grid, to a group of passive consumers. The production follows the demand. The electricity demand over the day on a large scale is highly predictable. The power production over the day is split up into segments: a base load which is available throughout the day, which is met by large, slow, efficient base load power plants (e.g. coal and nuclear power plants), load-following plants which run during the working day and the early evening (e.g. combined-cycle gas plants), and fast yet inefficient peak production plants for days with peak load (e.g. gas turbine plants). The use of power plants follows a merit order: the plants that are the least expensive to operate should run first. For security of supply, some plants are intentionally running while this is not directly necessary to meet demand. Both the grid and power plants prefer a flat demand profile; furthermore, power plants have ramp rate limits, i.e. their output should not change too quickly.

2.2.2. ENERGY TRANSITION

Conventional energy systems, including power systems, have various forms of pollution as side-effects, the most prominent of which is *carbon dioxide* (CO_2) emission. Furthermore, the fossil energy resources that fuel these systems are running out. To solve this, an energy transition has started, which aims to replace these

sources (and systems) with clean, renewable alternatives. The energy transition (Appendix A.1) disrupts the conventional operation of power systems. *Renewable energy sources* (RES) with an uncontrolled, variable feed-in profile, e.g. *photo-voltaic* (PV) and wind generation, are introduced at a large scale, which makes the load profile on the grid more volatile (Appendix A.1.1). Hereby, the grid needs more flexible generation; however, RES threaten the economic viability of the conventional power plants that can provide this flexibility. The energy transition also moves energy demand to the electric domain, which further increases the load on the grid (Appendix A.1.2). Consumers install *distributed generation* (DG) (Appendix A.1.3), usually based on RES, which gives synchronized feed-in peaks in the distribution grid. Storage (Appendix A.1.4) can avoid the use of flexible generation, by shifting production from periods with excess feed-in of renewable energy to periods with excess demand. Furthermore, local storage, or more generally *distributed energy resources* (DER) can reduce the load within the grid. However, storage resources are expensive, have limitations, and incur losses.

2.2.3. SMART ENERGY SYSTEMS

Smart energy systems (Appendix A.2) provide options to use energy more efficiently. We distinguish hybrid energy systems and smart grids.

2.2.3.1. HYBRID ENERGY SYSTEMS

Hybrid energy systems (Appendix A.2.1) exploit the properties of different energy sources, or synergy between different energy infrastructures. District heating (Appendix A.2.1.1) and *combined heat and power* (CHP) (Appendix A.2.1.2) are examples that exploit the synergy between different types of energy: by producing different types of energy at the same time, improvements to overall efficiency are possible.

2.2.3.2. SMART GRIDS

Smart grids (Appendix A.2.2) aim for similar efficiency improvements in relation to the use of the electricity grid. Typical goals of smart grids are load balancing, maintaining technical constraints in an affordable way, cooperation between connected entities on the grid, and acting on a power market for profit (Appendix A.2.2.1). We observe a split in the world of smart grids in two perspectives: a smart power hardware perspective and a smart coordination perspective. These perspectives are in principle orthogonal. We focus on the smart coordination perspective.

- 2.2.3.2.1. *Smart power hardware* (Appendix A.2.2.2) considers the more dynamic operation of the grid as a problem that should be solved by the grid operators and the power plant operators, for example with advanced power electronics and measurement equipment, storage facilities, better cables, and more flexible power plants. These improved components allow a business as usual approach to adapt to the immediate needs of the energy transition.

2.2.3.2.2. *Smart coordination* (Appendix A.2.2.3) views smart grids as a challenge of cooperation between stakeholders: by coordinating the needs of the grid operator and the users of the infrastructure, both can benefit. By this, the use of the production resources (including renewable generation), the grid infrastructure, and the DER can be improved from a global point of view.

Demand side management (DSM) has the potential to become a major source of flexibility for households, which may become much cheaper to buy and to operate than e.g. a battery (Appendix A.2.2.3.2). These resources are distributed throughout the grid, and can provide support for the grid operator at the specific point where it is most needed. DSM resources can be aggregated in a *virtual power plant* (VPP) (Appendix A.2.2.3.2), which can exploit the available flexibility on the market. Furthermore, DSM resources may perform dynamic curtailment when extreme grid conditions are detected (e.g. when the voltage or frequency is too high or too low).

As an example of how a bottom-up smart grid perspective can help to solve major challenges in the electricity system, we show the effect of the grid-aware control of a battery storage for PV (Appendix A.2.2.3.6). By optimizing for peak shaving rather than for self-consumption, the load on the grid decreases dramatically, while the amount of self-consumption remains the same.

DSM promises to bring cheap storage. However, this storage has some undesirable properties (Appendix A.2.2.3.2, Appendix A.2.2.3.3): in particular, due to user interaction, the available flexibility is not always known or well-defined. Therefore, if a controller does not account for this ambiguity, then it is often forced to take expensive repair steps. By this, DSM can give unreliable behaviour.

We review the properties of specific devices that are often considered for DSM (Appendix A.2.2.3.7). In particular the electricity consuming devices that are introduced by the electrification trend have a high potential for DSM.

Finally, we give a short overview of field experiments with smart grid coordination approaches (Appendix A.2.2.3.8), which demonstrate the feasibility of the concept in practice.

In the following sections, we consider the control, coordination, and optimization of these smart grid resources.

2.3. COORDINATION IN SMART GRIDS

Connected energy systems bring coordination challenges: given a set of resources that need to solve an energy balancing problem, the system should determine which resources should be used at what time, and in what way. If the resources are identical, then we can just pick some of the resources to solve the problem at hand. However, resources may be different in marginal cost, location, and operational constraints. The coordination should result in the *best* allocation of the available resources according to some objective or, given a bounded rationality in decision making, at least a reasonably good allocation.

These coordination challenges are not unique to smart energy systems. The large scale conventional energy systems already need coordination. However, smart energy systems offer many more control options than a conventional approach,

and thereby give much larger control problems. Furthermore, locality becomes increasingly important: the infrastructure can no longer be seen as a copper plate that transparently facilitates any desirable energy transaction, without regard to the cost or constraints of the infrastructure. Locality implies that a SES/smart grid control approach needs to account for these infrastructure aspects.

Next to the question of *which* resources are used, an important question is *when* resources are used. Most energy resources have dependencies over time (i.e. *state*): for example, large generators have long start up preparation times, high start up costs, and constraints on the ramp rate of the power output. There are also large differences between resources. A conventional coal power plant configured for baseload generation may take half a day to start generation, and another half a day to reach their full, nameplate capacity, i.e. rated maximum output [196]; in contrast, a flexible gas plant can ramp between minimum and maximum output within minutes [21], yet consumes more expensive fuel. DSM resources tend to have a fast response ability (within seconds to minutes), but have a limited availability and a constrained energy capacity.

The above discussion shows that state matters for generation resources, even though the reasoning about these resources usually focuses on marginal cost. For unconventional flexibility resources, such as DER, state is often an even more important aspect. For example, a battery needs to be charged before it can be used to discharge energy. By this, charging a battery on the one hand decreases the ability to store energy in the future, but on the other hand increases the ability to discharge energy in the future. Similarly, most DSM resources demand a fixed amount of “charge” (i.e. satisfy the demand) within a certain time period. In contrast, fossil fuel based generators can burn a virtually unlimited supply of fuel to give a variable amount of energy production, only limited by operational costs and constraints of the unit; however, there are examples where the fuel supply is constrained, as well as fuel supply contracts [168]. Summarizing, distributed energy resources and DSM do not only increase the scale of the control problem, but also increase the need to manage the state of the energy system. Also, the ability and availability of flexible generation to make up for variations in supply decreases,¹ and both DSM and RES introduce uncertainty. Finally, some problems have a sense of locality, where the use of a specific resource is preferred or even necessary (e.g. the self-consumption of PV). Consequently, uncertainty increasingly affects the operation of energy systems.

In the following, we discuss the coordination of energy systems. First, we address the conventional approaches to coordination in energy systems in Section 2.3.1. We continue in Section 2.3.2 with an overview of smart grid (and in particular DER) coordination. After this, we continue in Section 2.4 with an overview of the TRIANA DER coordination approach, which plays a central role in the remainder of this thesis.

1. Note that the conventional generators that *are* still available are increasingly operated in a flexible way at the cost of extra maintenance and lower efficiency, consider e.g. the references on flexible coal in Appendix A.1.1.

2.3.1. CONVENTIONAL COORDINATION

The smart grid aims to introduce coordination concepts to the distribution grid that are very similar to the common practice of the transmission grid: the operation of the transmission system has already been *smart* for decades. The “conventional” coordination of the grid seems to work well, even under substantial large shares of renewable generation. Therefore, we believe that it deserves attention as a source of design principles, and may give a starting point for coordination in a smart grid context. In spite of the large similarities, the smart grid domain has evolved virtually independent from the transmission domain.

2.3.1.1. PREDICTION AND PLANNING: BUSINESS AS USUAL

The operation of the current electricity system results from a large body of experience with the behaviour of the system, and the development of the system over time. The aggregate supply and demand for electricity follow predictable patterns, and are thereby amenable to forecasting. In turn, the forecast allows to plan the use of resources to optimize some objective (usually cost), and to ensure a stable system operation. The plans are constrained by the limitations of the infrastructure, and have to fulfill security constraints: if any one of the resources fails, this should never lead to a (large) system failure (i.e. an $n - 1$ criterion) [8,170]. Sufficient operating reserves need to be factored in to address these contingencies, as well as variations in system load. While the forecasts of the demand are accurate in general, nowadays large prediction errors (i.e. 20% of on line generation power) occur occasionally due to unexpected, synchronized RES conditions [255: p. 29]. As a consequence, keeping ample system reserves is essential. Reliability is an important consideration in smart grids as well, consider e.g. [129, 172] and the discussion on microgrids in Appendix A.2.2.1.3. Note that security in power systems usually refers to reliability/survivability, whereas the terms cyber security and smart grid security are used to refer to the ability to withstand *information and communication technology* (ICT)-based intrusion (see e.g. [347]).

Prediction and planning are performed on many time horizons. Long term planning (2 – 20 years) involves decisions about what infrastructure to build, based on expected increases in demand, and about the location of generation resources and load centers (expanding cities and industrial areas) [220]. As RES have location preferences (e.g. the wind is stronger at sea than on land, the sun has a higher intensity in Southern Europe than in Northern Europe, etc.), it is not always reasonable to build the generation resources as close as possible to the existing transport infrastructure or the load centers.

At the medium term (3 months – 2 years), decisions are made on the availability of the infrastructure and key generation resources, e.g. due to scheduled maintenance [291]. Some plants may be taken out of operation for months; the system operator should ensure that not too many plants are unavailable at the same time, in particular during periods of peak demand (e.g. winter in Central Europe).

At the short term, i.e. day-ahead horizon, operators decide how to use the available resources. These decisions usually consider multiple aspects: (short term) *unit commitment* (UC) decides which plants should be in operation during what

part of the day [283]; economic dispatch determines the generation profile of each plant [304]; and *optimal power flow* (OPF) accounts for the constraints and losses of the physical electricity network [234]. These aspects are strongly related, and therefore solved within the same, or a set of connected optimization problems.

At the intraday horizon (minutes to hours ahead), the units may be redispatched according to developing conditions (e.g. more or less wind than expected) [211], and balancing reserves are used to ensure a stable operation [39]. Usually, balancing reserves are split up into a tertiary reserve (15 minutes ahead, maximum 4 hours), secondary reserve (response within 5 minutes, maximum 15 minutes), and primary reserve (frequency regulated immediate response, 1% of on line generation capacity); the specific lengths and conditions are market specific [122]. As the time horizon decreases, the operation increasingly shifts from forecasting (what will happen) to observation (what has happened) and correction (how to deal with the observation).

At a very short time scale, i.e. the primary balancing, explicit coordination is no longer possible, or undesirable: at this time scale, we speak of (real-time) control. The definition of “very short” depends on the operator: while an automated system with electronic communication may coordinate within milliseconds, human operators communicating through telephones work at an order of magnitude of minutes. As it is critical for system stability, the primary balancing is generally implemented by *automatic generation control* (AGC), with strict real-time constraints and strong penalties for failures to respond, and capacity reservations [122]. The secondary reserve backs up the primary reserve, the tertiary backs up the secondary reserve, and redispatching may back up the tertiary reserve [221].

2.3.1.2. MARKETS

As mentioned in Appendix A.2.2.1.5, the energy markets have been deregulated over the last decades, which means that the resources and responsibilities of “the electricity company” have been allocated to separate parties. Where in the past only the connections across national borders and the negotiations with large consumers needed to be taken into account for electricity trading, the supply and demand of electricity are nowadays governed by markets and contracts. These markets emulate an optimization process, and should ideally result in a (near) optimal allocation of resources (that is, when the market design models the optimization process appropriately). The central principle of the market is competition by merit order: the cheapest providers can produce electricity, and the expensive resources will (almost) never be selected. The competition motivates investments in the resources, such that their operating cost becomes low enough to be selected, or motivates the decision to take the resource off the market (i.e. decommissioning). Similarly, consumers state how much they are willing to pay for their demand. Auctions are used to determine the balance between the supply and demand, and thereby the unit cost of electricity.

In the deregulated market, a *transmission system operator* (TSO) (US: *independent system operator* (ISO) or *regional transmission organization* (RTO)) has a monopoly and procures energy from market parties, using a regulated, transparent process (i.e. the TSO is the auctioneer, or delegates (part of) this responsibility

to others). The TSO is responsible for the *high voltage* (HV) grid and the operational system stability. Furthermore, the TSO manages the system *alternating current* (AC) frequency, which is a system-global property.

Similarly, *distribution system operators* (DSOs) own and operate the *medium voltage* (MV) and *low voltage* (LV) grids, and are responsible for maintaining service standards (e.g. voltage norms). Due to the passive nature of most of the resources of a DSO, the operational task in a conventional grid was in the past virtually nonexistent, except for various maintenance tasks and incident responses. The main decisions were in investments to adapt the capacity of the grid to changing circumstances. In the smart grid, the operational task becomes a lot more involved, although most of it will presumably be automated. The DSO manages the voltage, which is a local property. In an islanded grid, i.e. a microgrid, frequency management becomes a local property as well.

Next to the grid operators we have the *balance responsible parties* (BRPs), i.e. the producers, consumers, and traders on the market (see [315] for an overview of BRPs in the Netherlands). BRPs have contractual obligations for energy production and consumption. Smaller parties may outsource their responsibilities to a BRP. Consumers are usually aggregated and represented by an electricity retailer or an *energy service company* (ESCO)/aggregator [162]. In most cases, a BRP has (or controls) a portfolio of power plants and a retail division, which may be self-dispatched within the portfolio, provided that the TSO accommodates this [105, 294] (for example by auctioning transmission rights [175]).

A retailer (or aggregator) hides most details of the market for small consumers (e.g. < 1 MW). For residential consumers, flat per-kWh tariffs and flat *time of use* (TOU) tariffs are common. For commercial consumers, more complex tariffs schemes are used, which motivate a flat consumption profile without significant peaks in demand.

The markets to some extent mirror the temporal control hierarchy, i.e. they are split up in multiple time scales. Often, participation in the markets is mandatory for the larger producers and consumers [123, 365]. Furthermore, larger generators may be forced to contribute a part of the capacity to a service, e.g. the primary reserve [122]. The products on the markets for reserve capacity are described as *ancillary services*. These services usually have a payment both for the reservation and for the use of the reserve capacity [122]. Hereby, the payment for use of the capacity can be negative for down-regulation, considering the saved fuel costs in a thermal plant.

Over the last years, the short term markets have become increasingly popular, and are expected to become increasingly important under the influence of RES [163]. A short time in advance, the predictions of the production and consumption of electricity become more accurate and the realizations become (partially) known. In particular the day-ahead market has attracted attention, as it is long enough away for decision making (i.e. resources are not physically committed yet), and near enough to make meaningfully accurate predictions. Note that the day-ahead spot market, e.g. the APX market in the Netherlands [13], does *not* give “the prices” of electricity

for the coming day, but rather is the outcome, or clearing price, of an auction. More precisely, the prices describe the marginal cost of electricity at specific times of the day, that is, the cost of the most expensive producer and the least willing consumer.

The independent yet concurrent clearing of the market on e.g. hourly time intervals poses a significant risk to the participants: if a player bids for multiple time intervals and wins a set of noncontiguous time intervals, then it may be costly to supply the promised profile, e.g. due to extra cycling or discarding production. To avoid this, some markets accommodate the physical characteristics of the resources. The RES priority rules in e.g. Germany are an example of this. Energy is not only sold on the day-ahead market, but also at longer time scales, by which these discrete aspects can be amortized. Alternatively, some markets support structured bids. A simple variant of the structured bid is the block bid, which proposes supply or demand in a set of predefined blocks in the day [14], or in consecutive time intervals (e.g. the APX and NordPool offer consecutive blocks and linked blocks, and the APX also offers mutually exclusive block bids [13, 245]). Some markets offer more complex bid structures, which for example allow players to model start-up costs and ramping constraints [8, 65, 310].

2.3.1.3. OPTIMIZATION

Large energy resources, such as power plants and the transmission grid, give large opportunities: even small gains – for example 1% lower losses – translate to savings that are meaningful at a national level. For example, the Netherlands consumed 109 TWh of electricity, of which 4.4 TWh are grid losses, at an average market price of €50/MWh [89: pp. 44, 56]. Not accounting for the coincidence of these losses with periods of high demand, these losses cost at least €220 000 000. A 1% reduction within the losses would save €2.2 million per year. Therefore, a lot of effort has been invested (and continues to be invested) into making better use of these large energy resources. In Western countries, many aspects of large scale energy resources, and the transmission and generation system in particular, are subject to optimization, often with manual tuning by an operator [150]. In the following, we present an overview of the large body of work on the operational optimization of large scale power systems.

- 2.3.1.3.1. *Centralized optimization* In the second half of the 20th century, various optimization methods have been developed for the national electricity systems, which often follow a centralized design [234]. Even though these systems are relatively small (e.g. 200 units [228]) in comparison to the ambition of the smart grid, a common theme is that the optimization problems are too large to be handled in a monolithic manner. To address these problems, various approaches have been used to split up optimization problems, which is more formally described as *decomposition*. Decomposition gives a top-level master problem and a set of subproblems, which are solved separately and subsequently communicate their outcomes to the master problem. Note that the original problem is usually never represented explicitly, i.e. the procedures more resemble a composition approach in practice. These optimization methods often need domain specific heuristics to give fast convergence. Just as in other fields where optimization is applied, *linear programming* (LP) and

quadratic programming (QP) based methods are popular in this domain [235], as well as *nonlinear programming* (NLP) techniques [234] (Bertsimas [29] provides a background on linear programming).

At EDF (the French electricity company), a Lagrangian relaxation based method has been used for unit commitment [228]. This method separates the system into a global problem that describes “coupling” constraints (i.e. system balance and security constraints), and a set of local problems that describe the local constraints (one for each unit). Lagrangian relaxation expresses the coupling constraints as penalty terms in the objective function. The rewritten constraints have associated weights, or *Lagrange multipliers*, that are used to represent a constraint in each of the local problems. The local problems themselves are solved by specific *dynamic programming* (DP) formulations that account for various local constraints, e.g. minimum up-times and down-times. The global optimization now involves finding a set of Lagrange multipliers, such that all the given global constraints are (almost) met. Whereas finding these multipliers may be time-consuming in general, a problem specific price update heuristic helps to direct the search. For the practical example in the paper, a near-optimal solution is found in 28 iterations (2 minutes on a mainframe in 1983). Note that while these optimization methods gave considerable (i.e. mainframe scale) computational challenges in the 1970s and 1980s, these problems can nowadays be solved on basic hardware.

As computing power and optimization techniques improve, solution techniques that consider more aspects at the same time are nowadays again considered, such as *mixed integer (linear) programming* (MIP). In addition to an LP, a MIP can take into account discrete aspects, such as unit start-up behaviour and up/down time constraints. In [207, 310], a comparison is presented between a Lagrangian relaxation and a MIP approach, in the context of the implementation of a MIP based approach (at PJM, a large RTO in the US). The MIP allows for easier modelling with less domain specific knowledge than a Lagrangian relaxation approach, and is not susceptible to local minima. However, MIPs do give an exponential (and occasionally erratic) growth in solution time and memory use. The solution time depends strongly on the chosen problem formulation; therefore, for industrial-scale problems, significant gains are possible by spending effort on model tuning and reformulation [167, 293] (note that this is similar to e.g. software programming and hardware synthesis, where a difference in problem formulation or tooling configuration can have a major impact on the execution time or circuit size, which is incomprehensible for users without knowledge of the underlying mechanism). The problems are often hard to decompose effectively, unless the problem has suitable properties [212].

- 2.3.1.3.2. *Market optimization* Deregulation gives an extra dimension to the optimization problems. It leads to the creation of markets, which changes the nature of optimization problems. Now aspects of competition come up and introduce game theoretical components into the optimization. Rather than trying to find a global optimum, each player on the market tries to optimize against the market model; in a good market design, these should be near-equal. Instead of optimizing for system security, market participants optimize against imbalance penalties. Competitive behaviour

may lead to inefficiencies: for example, there are incentives to misstate the parameters of resources (e.g. to convince the auctioneer not to turn off a generator) [299], in particular in the presence of nodal pricing [149]. Furthermore, when a player has market power, it can manipulate the system for its own benefit. Also, opportunities for cooperation between stakeholders may be ignored or even forbidden [247].

Therefore, a market paradigm gives two important challenges: market mechanism design, and the development of strategies for these markets [338]. The market mechanism design is usually taken as a given (Section 2.3.1.2). However, note that the market mechanism is far from static in practice, as it is nowadays continually adapted in response to undesirable behaviour from market players (e.g. changes in feed-in rules and tariffs). Since the deregulation, bidding strategies on electricity markets have been a popular (and economically lucrative) research topic. Players generally do not model the behaviour of other players explicitly, but instead use models of the uncertainty of relevant market parameters, e.g. the clearing price and demand; for a review on optimization based bidding, see [200].

2.3.1.3.3. *Uncertainty in optimization* Uncertainty is a daily fact of life in power systems operations. However, in large scale power systems security constraints also play a crucial role. Combining these two poses a significant scheduling challenge on the operation of power systems. The conventional approach to address uncertainty is overdimensioning according to an $n - 1$ criterion (or $n - x$, for some value of x): the optimization problem ensures that at all times, enough units are committed and enough spare line capacity is available to make up for the loss of any single (or x) unit(s) or line(s). To avoid the complexity of considering every combination of faults, conservative estimates are used, e.g. using the highest committed plant to determine the necessary reserve, and modelling only critical parts of the infrastructure explicitly (e.g. a single line between islands [8]). Note that this approach implies that units should be similar in size, or at least smaller than the largest unit, to avoid excessive reserve requirements; in practice, this limit is e.g. 1 GW, depending on the system size and already installed units.

As resource overdimensioning is expensive, less conservative scheduling methods are considered that account for the likelihood of realizations. Stochastic programming methods are mainly used to solve these problems. A common technique to model stochastic behaviour is by the use of scenarios, which can be represented in a scenario tree. Each of the branches has an associated probability weight that represents the probability that a specific scenario occurs. In spite of the stochastic simplification (in comparison to a complete game-theoretic view on the market), this still leads to large scale optimization problems [338]. To solve these problems in reasonable time, approximate solution techniques are necessary, e.g. scenario reduction techniques [142]. Alternatively, (meta-)heuristic and *artificial intelligence* (AI) based optimization approaches (e.g. evolutionary and learning algorithms) are prevalent, which try to find a control policy to work in the given environment [351]. As we present an optimization approach based on metaheuristics for DSM in Chapter 4, we continue this discussion in Section 4.3 (and refer to [35] for a literature review).

In many stochastic optimization problems, some decisions can be revised during the operation of the system: over time the realization of the uncertainty is revealed, which allows for a “wait-and-see” decision model [70]. For example, whereas the decision to turn on a power plant has to be taken far in advance, the actual output can be changed during operation. These type of problems are described as multi-stage problems. A specific well-studied subclass of these problems are two-stage problems with recourse [181]. For some of these problems, decomposition schemes are available, which allow fairly complex, industrial-size problems to be solved, if the number of scenarios is small enough or sufficient parallel computation capacity is available [300, 368].

As already mentioned, representing the complete probability distribution of the uncertainty in an accurate yet compact way is a difficult problem, in particular if there is a large number of unlikely yet high impact events (e.g. for the transmission grid, any combination of lines and generators can fail for any duration of time, leading to an intractably large number of scenarios to consider for an acceptable coverage). Robust optimization presents an alternative approach to stochastic modelling [26]. Rather than optimizing for a specific probability distribution, robust optimization satisfies the constraints under all realizations within an uncertainty set. The set models all “reasonable” instances of the uncertain variables that need to be covered by the solution. Intuitively, the set aims to cover a certain probability weight (e.g. 99.99%; however, since the true probability distributions are not known, reliability claims are usually not made in terms of probability weight). A robust optimization problem may be viewed as a game, where the operator aims to minimize cost, and the environment subsequently tries to maximize cost (or better: to maximize damage) under the chosen policy. Rather than allowing any possible deviation, the environment has a maximum “budget of uncertainty”. Analog to the stochastic approach, in some problems the operator is able to respond to the behaviour of the environment. If the uncertainty set has a suitable structure, then efficient solution methods are available [26]. Robust optimization is amenable to various decomposition methods, which allows large problems to be solved [28, 375]. In terms of conservativeness, robust optimization roughly falls between the worst-case and stochastic approaches, where the level and direction of conservativeness is determined by the size and shape of the uncertainty set [28]. Robust optimization is considered in several papers on power system optimization [27, 220, 305], where it gives promising results at a much lower computational cost than stochastic optimization: for example, Bertsimas [27] solves an industrial-scale security constrained unit commitment problem with 312 units on a laptop PC in 6 hours, whereas the original problem with conservative (reserve adjusting) security constraints took 1½ hours (by increasing tolerances, the solution time for the original problem can be matched). A comparable large problem with 169 units and 12 scenarios is solved with stochastic programming in 30 hours [368].

2.3.1.4. CONTROL OF DISTRIBUTED ENERGY RESOURCES

The optimization of production resources relies on knowledge (i.e. estimates) of the consumption. In the conventional context, retailers already coordinate DER, and in particular DR resources, to improve the shape of their demand profile, although much more coarsely than smart grids propose. In [141, 286] and related work, DER control is categorized in four categories: passive control, active control, interactive control, and transactive control. We use the first three categories to describe the control methods that are used in current practice for the control of DER and in particular DR resources. We defer the discussion on transactive control to Section 2.3.2.

The most common form of “coordination” is passive control. For example, *time of use* (TOU) pricing schemes motivates customers to consume electricity e.g. at night, which raises the baseload and decreases the peak load on the grid during daytime. Many devices that are suitable candidates for DR already have the ability to delay their execution until the start of the low tariff, or “pre-charge” before the start of the high tariff. Similar systems are used to address *real time pricing* (RTP)² and *critical peak pricing* (CPP) schemes [116]. These “dynamic” pricing schemes fall in the category of active control: they rely on information provided by the operator, rather than a static configuration. These values and events are often communicated through the smart meter, as the meter also has to account for the change in tariff (similarly, legacy meters rely on a pulse code superimposed on the grid frequency to switch between tariffs). Based on experience, the retailer predicts the (both manual and automated) response of the cluster on the price signal (vector), which it can then use to plan the use of its other (e.g. production) resources. CPP schemes usually have a limit on the number of invocations on a yearly basis (e.g. 10 – 20 times); the operator therefore needs to conserve this resource [263]. A variant of CPP is *critical peak pricing with rebate* (CPR) (described in [116] as *peak time rebate* (PTR)), which gives a rebate for consumption below a baseline consumption level. Although this variant is popular in practice, it has some problems, e.g. that the baseline level can be effectively gamed, and that it motivates varying demand [38].

Whereas this approach has proven to have effect on the demand over the last decades [42] and promises to bring dramatic savings (billions of euros per year in the *European Union* (EU) [116]), it has several shortcomings in the context of the energy transition. The approach offers coarse control over the demand profile: it is designed as a supplement to relieve stress on the supply and the grid, rather than as an active system operation instrument. Furthermore, it assumes that the requested response is fairly static in nature, e.g. to shift demand to the night. However, in an energy transition context, the desired response depends on the type of day (e.g. sunny, windy, or neither), and should be flexible to respond to RES variations.

2. In embedded systems, “real-time” has a very specific meaning [52] that does not correspond to RTP: the prices are generally updated on a daily basis, making it more accurate to describe RTP as a day-ahead n -tiered TOU approach (usually with $n = 24$ or 96).

To address variability and to increase the precision of control, some types of DR allow an operator to dynamically increase and decrease load (see Appendix A.2.2.3.2). These approaches fall in the category of interactive control, where the operator can measure the response, or the resource reports its response. Just as in a static approach, in terms of coordination, the operator should account for rebound effects [256] and response fatigue [187], i.e. it needs a model of the behaviour of the resources. For control, the operator may use grid or radio pulse codes to “release” a demand group with a known approximate behaviour, e.g. 6 groups of 5000 boilers [189]. Similarly, an operator with more fine grained access can perform *direct load control* (DLC) on specific devices, e.g. by GPRS, *power line communication* (PLC), or using the smart meter [148, 346, 295: pp. 473–474]. Note that at this point, the difference between “conventional” and a “smart grid” coordination becomes a grey area.

A different type of coordination that is passive in the sense of information exchange relies on the physical properties of the grid. For example, modern PV inverters respond to violations of the AC voltage and frequency norms to avoid excess feed-in. In spite of a lack of information exchange, an inappropriate control can still lead to system stability problems: in Germany, all DGs, and in particular PV inverters were configured according to law to shut down at a threshold frequency of exactly 50.2 Hz. This led to a major European transmission level event in 2006, and a subsequent retrofit operation of 315 000 PV plants at an estimated cost of €175 million [12], with an ongoing operation to reduce the number of affected plants [83]. Nowadays, randomized cut-off frequencies and droop control are used, depending on the size of the plant [73, 324]. Similarly, prototypes of “grid friendly appliances” have been developed that dynamically respond to the voltage and/or frequency in the grid, e.g. dryers and EVs [147, 274]. Whereas these physical control strategies directly address standard violations with associated safety risks, these protection mechanisms are not designed to give deliberate economic behaviour. Also, as illustrated by the PV frequency issue, physical characteristics are synchronized, and require careful deliberate randomization to avoid introducing new stability problems.

2.3.2. SMART GRID COORDINATION

Within smart grids, DER are considered as one of the main energy resources, in contrast to the view in Appendix A.2.2.3.2 that puts DER/DR under the direct control of an operator. According to the categories of [286] (see Section 2.3.1.4), this corresponds to transactive control, or indirect control in [152]: there is not just control of devices, but also a negotiation process to determine which resources should be used. Due to the large number of devices and the envisioned time scale, this process needs to be automated, at least from the operator side. From the user perspective, a large degree of automation is desirable as well (see Appendix A.2.2.3.2). The negotiation process is usually based on some type of steering or optimization approach.

In comparison to the conventional context of Section 2.3.1, the smart grid control literature has not yet converged to a clear common conceptual framework (e.g. with concepts such as unit commitment, economic dispatch, and security con-

strained OPF); however, some common patterns have started to emerge over the last years, which we discuss in this section. Vardakas [335] attempts to survey this literature, which is hard to follow at the current rate of publication (see [335: Fig. 2]). In this work, we aim to maintain structure by relating the work to conventional power systems and to well-known, accepted smart grid concepts.

In smart grid problems, the system objective is usually defined by the environment: it generally includes a subset of the goals from Appendix A.2.2.1, where the choice depends on the stakeholders that are involved. Furthermore, multiple “smart grid providers” or aggregators may compete for the same resources [74, 75].

2.3.2.1. COORDINATION BY OPTIMIZATION

The coordination of DER uses techniques from the fields of mathematical optimization, and therefore considers similar approaches as the conventional problem. In the following, we discuss several approaches for coordination approaches that are based on optimization. We first address centralized optimization approaches. Next, we present decentralized optimization approaches, which relate to the approach that we follow in this work. We subsequently present direct market optimization as a variant of decentralized optimization. After this, we discuss aggregate modelling approaches. Finally, we discuss control based on the single commodity Walrasian auction, which is prevalent in smart grid literature.

2.3.2.1.1. Centralized optimization The most straightforward way to coordinate a group of devices is to gather the complete problem structure at a central point, make an optimized schedule for these devices, and distribute the schedule. In this context, MIP and *mixed integer quadratic programming* (MIQP) formulations are popular, because these give optimal solutions (in relation to the problem formulation) with a limited modelling effort, provided that the system is not too large or complicated. Conceptually, this is similar to the conventional power system day-ahead planning (unit commitment) problems, where the system operator determines when each of the large power plants should run (Section 2.3.1).

Centralized methods give a clear overview of the state of a system, and the (expected) interactions between components. However, the scalability is often an issue: whereas the centralized optimization of the energy demand in a single household or the day-ahead scheduling of large generators are feasible, a centralized design limits the system size and gives a large control latency. From a computational perspective, these limits are addressed in part by heuristic optimization and decomposition techniques, optionally in combination with parallel processing. The communication to the central controller tends to become a performance bottleneck, as the controller needs to receive the state of each device, and send commands to every device. Furthermore, a central component (without redundancy) introduces a single point of failure: the coordinator, and the communication to the coordinator must always work. Hereby, we believe that an explicit centralized optimization approach is inappropriate in the context of the distributed, large scale smart grid problems that we aim to address.

2.3.2.1.2. *Decentralized optimization* The control of a large smart grid leads to large optimization problems. To tackle these large problem sizes, a practical solution strategy needs to be able to split up optimization work. For scalability, the work should be distributed as well. In decentralized optimization and control for smart grids, decomposition is a central principle, just as it is in large scale conventional power system optimization problems (see Section 2.3.1.3.1). The difference here is that (part of) the optimization of the subproblems is performed locally, rather than on a mainframe or a computing cluster. Hereby, (limits on) communication and local processing workload become relevant. In this case, the communication concerns the coordination mechanism, rather than the internal problem structure and direct control commands. According to the classification in Section 2.3.2, this is a form of indirect or transactive control rather than direct control.

In many decentralized configurations, the local subproblems are able to function independent from the coordinator, although with degraded performance. By this, a decentralized system architecture can improve reliability.

» *Natural structure* The hierarchical and usually radial tree structure (i.e. only top-down lines, without loops) of the LV distribution grid gives a convenient, decoupled structure for decomposition: the household control optimization problems are virtually independent from each other, except for their contribution to the load on the grid (the “tree” structure is in fact a *directed acyclic graph* (DAG) if the internal structure of an LV distribution feeder is considered, which carries multiple, separate AC phases; for simplicity, we consider it as a tree with bundles of edges; current smart grid optimization approaches usually ignore these technicalities, and consider only aggregate electric load at the top of the hierarchical level). Many hierarchical approaches can be nested: the optimization problem may be split off at the edges in the tree; the subtree becomes a separate subproblem. In a MV distribution grid, loops are common, which complicates splitting techniques (loops make physical flows harder to predict); however, the lower-level (below-transformer, LV) problems can still be considered separately. Similarly, most device control optimization problems within a building are only coupled by their common use of the grid connection (however, most thermal systems pose an exception to this rule).

At the bottom of the hierarchy, a (relatively small) optimization problem is solved as a single block. The bottom problem may be a centralized building optimization problem, or a device optimization problem. The decomposition method may rely on certain properties of the underlying problem and the solution method, e.g. linearity or convexity/concavity.

» *Dual decomposition* Lagrangian relaxation, or dual decomposition [257, 276] is popular in optimization and control, both in theory and practice. These methods represent the (violation of) constraints in the objective value.³ If the master problem and the subproblems have a suitable structure, then these methods have

3. Note that dual refers to duality rather than having two subproblems, although most examples do illustrate the concept with two connected problems for simplicity.

provable (but not necessarily fast) convergence properties [133]. Dual decomposition methods are well known in conventional power systems optimization (Section 2.3.1.3.1), and also in smart grid control, e.g. [32, 282]. During the optimization process, the dual solution gives a lower bound on the primal (i.e. real) optimal solution, and may thereby help to quantify the quality of a found solution. To search for solutions, many variants of (sub)gradient methods are used [41, 378]. These methods generally rely on a reasonable smoothness of the master problem and the subproblems, to allow a “gradual” convergence towards the optimal solution.

A related set of methods that improve convergence are the augmented Lagrangian methods [88] and proximal methods [258], e.g. the *alternating direction method of multipliers* (ADMM), which are used for smart grid problems as well [69, 145]. Although these methods often still have a very long “tail” towards the optimum, adequate solutions are found in a reasonable number (e.g. 10 – 100) of iterations for many problems; consider e.g. [258: Figure 7.1].

» *LP decomposition* For LP problems, several decomposition techniques are available; (delayed) column generation, and constraint (i.e. row) generation approaches (and combinations of both) are of interest to this discussion. Both approaches exploit a “diagonal blocks” structure in the LP coefficient matrix to avoid representing (and solving) the complete problem in one go. These structures naturally arise from a structured introduction of the subproblems in the formulation, i.e. a subproblem has a “private” set of variables and constraints, with a narrow connection to the master problem. Instead of a direct solution, the subproblems are consulted iteratively to construct a master problem that exactly represents the original problem, but may be orders of magnitude smaller (or even approximate a model of infinite size).

Dantzig–Wolfe decomposition, originally known as *the* decomposition algorithm for linear programs [71], is a column generation algorithm. The algorithm works for problems with an LP matrix structure with the aforementioned “diagonal block” structure, which are subsequently connected through only a small number of connecting constraints (e.g. a power balance equation). Using the LP (or in fact, the solution of the dual of the LP problem), column generation can determine the “marginal cost” (or *shadow prices*) of the constraints. By this, it can systematically construct the subproblem solutions that are able to improve the objective. Column generation is closely related to Lagrangian relaxation, yet exploits the linear structure of the problem [214]. Column generation is applied in recent smart grid literature, e.g. [HT:3, 225, 226, 303, 306],⁴ and in [40: p. 91]. We address column generation in more detail in Section 3.4.

Constraint generation is the dual of column generation, and relies on a similar matrix structure. However, constraint generation assumes a small set of connecting variables rather than connecting constraints. Instead of constructing candidate solutions for each of the subproblems, the subproblems generate “counterexamples” for a variable assignment that is proposed by the master problem, or more formally

4. Incidentally, each of these column generation approaches for smart grids were presented on the same conference in 2013.

cuts. Benders decomposition, or Benders cut [25, 127], is a well known procedure to generate these cuts. The procedure uses “L-shaped” matrices, i.e. (in the generalized form) a block diagonal structure with a small number of shared variables on the left-hand side of the matrix [56, 300]. This approach is very common in power systems optimization, due to the ability to effectively handle security constraints: the number of constraints is in principle very large, but only a small number of these constraints are relevant in the optimal solution [289, 290]. In two-stage stochastic optimization with recourse, the approach is used to give a common variable assignment for the first stage problem, where each subproblem represents a realization of the recourse on a scenario [300]. In robust optimization, a single “adversarial subproblem” determines the worst reasonable scenario, considering both the uncertainty budget and available recourse [296, 318]. Literature applies Benders decomposition in various smart grid control optimization problems, e.g. [185, 186, 292, 374]. Column and constraint generation may be combined with a nested problem structure [375].

2.3.2.1.3. *Direct market optimization* Some decentralized methods do not introduce an intermediary coordination mechanism, but rather target a given market directly: e.g. at a building or household level, DER are applied directly to a market scheme without cooperation with a coordinator. From an optimization perspective, these are usually centralized by design, and follow similar (although often smaller-scale) optimization techniques as the centralized optimization approach, although the focus is often more practical (e.g. from the perspective of a building owner). Many different methods are used in practice, see e.g. [295: p. 470], including mathematical programming/MIP models [222], set-point learning control [121], and device-level DP [366]. Most current commercial smart energy products and *energy management systems* (EMS) fall in this category, including the residential storage solutions as discussed in Appendix A.2.2.3.6.

2.3.2.1.4. *Aggregate modelling* An emerging category in smart grid optimization is aggregate modelling, or *aggregate and dispatch* [74]. Motivated by the difficulty to accurately model and optimize individual resources, these approaches represent groups of resources as a single entity, similar to the way that demand forecasting considers large groups of devices. However, whereas demand forecasting only predicts the behaviour, the models considered here express and exploit the available flexibility in behaviour. A dispatcher determines the specific allocation of the flexibility, usually dynamically (“in real time”/just-in-time). Hereby, aggregate modelling represents a hybrid between top-down and a bottom-up load control, or an *implicit* centralized optimization approach.

The aggregate model exploits that the “units” (i.e. devices) have a different sense of commitment than the large thermal plants. Whereas large thermal units on the one hand need to be individually prepared (and thereby committed) for operation in advance, i.e. by carefully heating it up for hours, most DER can be dispatched dynamically. On the other hand, DER units are committed after dispatch, and switching actions should be limited. Furthermore, especially for DR resources, the dispatch still needs to specify which units should be committed: an aggregate

FIGURE 2.1: Example of constraint aggregation in an energy flexibility space for a group of EVs with common arrival times and deadlines [331]. In this example, the optimization chooses a path through the flexibility space that starts charging after 2 hours and completes after 10 hours. The upper figure shows the corresponding power profile of the aggregate consumption plan.

view does allow the dispatcher to wait and see which units become available, but still needs to account for local constraints, e.g. specific heat buffers have to be filled at certain times, and specific EVs need to be charged before their charging deadline.

The Intelligator [156, 206] is an example of such a control approach, which uses a bottom-up energy flexibility aggregation, which is called a *three step approach* (constraint aggregation, optimization, and dispatch) [331]. As an alternative to explicit aggregation, Claessens [61, 62] uses reinforcement learning on this abstract model (i.e. with the minimum power, maximum power, accumulated minimum energy consumption, and accumulated maximum energy consumption over time). The learning is implemented at a cluster level rather than at the level of individual devices, which makes prediction less difficult. Figure 2.1 shows an example of an aggregated flexibility space, together with a possible planning within this space. We use the same model in Appendix B.6 to determine a lower bound for a large optimization problem. To address uncertainty, a stochastic programming approach based on *approximate dynamic programming* (ADP) is used [275]. The approach uses a priority scheme to dispatch the resources, which is implemented with a double-sided Walrasian auction (which we discuss later in Section 2.3.2.1.5). In [74, 165], the bidding functions include a voltage droop control, which addresses local voltage events dynamically.

Biegel [31] proposes a comparable top-down load control approach to control a portfolio of heat pumps that estimates the energy storage capacity and power consumption limits at a cluster level. A forecasting and *model predictive control* (MPC) at a cluster level reasons about the behaviour of the cluster over time, according to

the “global *state of charge* (SoC)” and the estimated minimum and maximum power consumption of the cluster. Individual devices only report their (normalized) SoC and on/off state, by which the dispatcher can determine the order in which devices should be started based on a sorting algorithm (which are very efficient, i.e. typically $\mathcal{O}(\log n)$ per device in a group of n devices) [30].

Koch [193] discusses an aggregated model for the demand of thermostatically controlled loads, which sets up a Markov model of the state distribution of the loads; the state consists of the (discretized) SoC and the mode of the load (on or off). The central controller can broadcast a request for loads in specific state bins to switch on or off with a given probability, which changes the state distribution (i.e. some devices go from the “on” to the “off” bin and vice versa, which changes their SoC travel direction). A small sample of the loads reports its state to the central controller, which gives an estimate of the global state. Based on experience with the response behaviour of the cluster, the controller can perform MPC using a Markov model of the behaviour over time.

A aggregate approach is to some extent robust against mispredictions of individual resources, by dispatching resources dynamically. However, by lumping, the planning loses local information [31]; some of this information is reintroduced by the dispatch (e.g. voltage limits [350]). Discrepancies between the aggregate model and the dispatch model lead to path deviations in practice [74]. The individual resources still need to be aware of the local available flexibility (e.g. when a buffer needs to be filled), and thereby need a prediction of the expected local load. However, in comparison to an explicit planning approach, the aggregator can wait to see which specific resources become available, and estimating the cost of prediction errors (e.g. extra auxiliary heating) as a group may be less difficult. Furthermore, it is still sensitive to correlated disturbances, e.g. differences in aggregate load and PV feed-in, which may be addressed on a cluster level [275].

2.3.2.1.5. Single commodity Walrasian auction A specific type of decentralized optimization that is prevalent in smart grid DER control [124, 195, 206, 381], is based on the two-commodity Walrasian auction [345], which is interpreted as an single commodity auction between supply and demand. The PowerMatcher is an example of a system that uses the Walrasian auction as the main concept of control [195]. Walrasian auctions describe an economic concept of supply and demand matching, where a group of producers and consumers can exchange a good under the supervision of an auctioneer under perfect competition. The good (commodity) has a price per unit: if the price of the commodity increases, then producers are willing to incur higher costs to produce more units. With such an increasing price, the demand of consumers for the commodity decreases. The auctioneer now has to find prices such that the supply and demand for all commodities are balanced.

If there is a single commodity (or the prices for the other commodities are fixed), then the supply of each producer can be given by a supply offer curve, and the demand of each consumer can be given by a demand bidding curve. A supply

FIGURE 2.2: Supply and demand matching by a Walrasian double-sided auction for an electricity market. The illustrated one-commodity market clears at a price of €0.22/kWh with a clearing quantity of 600 kW.

offer curve is nondecreasing in price (i.e. a higher price per unit implies the same or a higher supply), and a demand bidding curve is nonincreasing in price. The supply and demand curves of all individual producers and consumers can be added up to find an aggregate supply and demand curve, respectively. At some price, the quantity of supply and of demand meet, i.e. the market clears, which leads to the *equilibrium quantity* and *equilibrium price*. The equilibrium price determines the value of the commodity: at this price, the marginal cost for production and consumption are equal, which gives an economic optimal behaviour, given that the market conditions are satisfied. After the auction, the price can be used to allocate the production and consumption quantities to the individual players on the market, according to their bids.

Figure 2.2 gives an example of supply and demand matching by a Walrasian auction for a small electricity market. In this figure, the price is placed on the horizontal axis. The market allocates a set of flexible generators for a set of flexible loads. The supply curve in this example may result from the aggregate output of four 200 kW generators, one of which is of a “must-run” type that runs regardless of the price. Each of the flexible generators has an (economic) minimum production of 50 kW and a variable output of 150 kW; the generators have a different price response range. In practice, a variable price response results from the use of a different fuel, or other operational expenses. At the demand side, an increasing cost leads to a decrease in demand for electricity. The market clears at $\approx$ €0.22/kWh, for a supply of 600 kW. The supply is provided by the generators with the lowest cost (all except for the most expensive one in this example).

The supply and demand bidding curves give a way to reason about the roles of different energy resources: not all types of flexibility are equal. DSM aims to make the demand more flexible, i.e. increase the vertical range, and shift the demand curve to the left. Industrial, large-scale DR mostly adds flexibility at the left-hand side for cheap production. Residential DR (i.e. the current practice in the US) adds flexibility on the right-hand side, avoiding consumption during critical peak hours.

Hybrid energy systems add flexibility at the left-hand side to exploit cheap electricity (e.g. heating rods in a self-consumption application), or at the right-hand side to avoid excessive cost (e.g. *plug-in hybrid EVs* (PHEVs)), depending on the suitability of using electricity for the process at hand (using electricity for a heating rod is generally less efficient than heating with gas, and charging a PHEV with electricity is generally more efficient than running it on gas). Base load plants and renewables push the supply curve upward, which decreases the price of energy; subsequently, the expensive units at the top right will (almost) never run, and are therefore taken off the market. Renewable curtailment lowers the bottom left of the supply curve. Making plants more flexible increases the vertical range, but moves the supply curve to the right. Storage couples the supply and/or demand curves over time, which increases the top right of the supply curve and the top left of the demand curve.

Single commodity Walrasian auctions have various useful properties. As already illustrated, the auctioneer can rationally reason about aggregates of producers and consumers, and allocate production and consumption accordingly. The bidding curve can be made explicit, or sampled from the players in a structured way (*tâtonnement*), i.e. by binary search. Explicit bidding curve representations can use efficient, composable representations, e.g. piecewise linear functions that can be merged or consulted as a tree [369: p. 57], or use binning [259]. The bidding curves give information about the minimum and maximum possible supply and demand quantities, and give an indication of the price sensitivity across the demand–supply curves.

However, single commodity Walrasian auctions also have drawbacks. By definition, these only optimize for a single commodity. Whereas Walrasian auctions with multiple commodities are possible, these have less useful properties and are more difficult to solve than single commodity auctions; see for example references [16, 343, 352, 369]. The structure of bids imposes restrictions on the possible control actions that can be offered: the ascending (or descending) bid order forces players to expose only the best bids, which can make market clearing more difficult when there are step changes in the bidding functions (e.g. when a generator’s cost per unit decreases with the volume, it should bid only for zero production and for full production). Furthermore, the auction assumes zero exchange costs and perfect competition. Locality violates both of these assumptions: reducing the costs of energy exchange is one of the main objectives that smart grids aim to achieve, and there is a strong preference for the use of local resources, which implies an unfair competition from the Walrasian perspective.

A Walrasian auction is designed to match production and consumption, according to their marginal cost and value. In contrast to a centralized paradigm, prices are not the main consideration in a distributed context: a smart grid has more objectives, and constraints. With transformations of the supply and demand curves, many objectives and constraints can be represented. For example, an auctioneer can impose transport costs, and impose limits on both individual and aggregate supply and demand. In the electricity domain, a subset of transformations is known as nodal (or zonal, or locational marginal) pricing, which is used to address limited corridors in the transmission grid; see e.g. [260]. However, while these transfor-

mations may give practically useful allocations, the economic principles of the auction are lost: it is not clear which player accounts for the difference in price (and thereby cost) between production and consumption. Usually, in the transmission grid, the chosen solution is to socialize these costs, or to auction the capacity on congested links separately [260].

In energy systems with storage, shifting supply and demand over time is a central feature. Single commodity auctions may be used to address time by holding a sequence of auctions that address successive time intervals, or by holding a simultaneous auction for all time intervals, with separate bids on each market. In essence, time introduces a special type of a multiple commodity problem [17]. Storage (or more generally: state) introduces dependencies between time intervals. As the markets are not cleared together, players have to speculate on the bidding curves of other players at later time intervals.

To obtain the desired response from a group of players with stateful resources, it is necessary to define and tune a bidding strategy for each player that is embedded in a (software) agent [195: p. 169]. The bidding strategy should consider not only the expected clearing prices in the future, but also the state and behaviour of the agent itself. For example, a battery runs out of flexibility at the lower and upper bound of the SoC. More pressingly, most DSM resources must run before a certain time. Therefore, the bidding functions of device agents do not only consider an estimate of profit, but also incorporate an element of a risk strategy: as the deadline to satisfy the demand approaches, the bid price to satisfy the demand increases. The shape of the curve over time determines the device's risk profile. At the end of the flexibility range, the bid price may increase hyperbolically to reflect that it becomes a must-run load [206]. The control can exhaust the flexibility of *all* devices in the cluster in at least one direction, which is described as *incentive clipping* [355]. To prevent these scenarios, agent developers set up conservative policies, e.g. trying to charge EVs as soon as possible, while remaining below the rated power of the transformer [130]. In these applications, the auction works essentially as a dispatch mechanism, where the price serves as a relative priority rather than as an economic objective. A similar but gradual "fatigue" response occurs in [354], although a synchronized breakdown is not observed in the presented scenario. These examples demonstrate that the coordinated control of large groups of DER, whereby a synchronized cluster of devices can achieve a high ramp rate, is not only a benefit, but also an operational hazard when the cluster fails as a whole (note that heterogeneous thermostatically controlled loads are exhausted more gradually, as the end of the SoC range is generally not reached at the same time). In [195: p. 171], device flexibility is characterized as a continuum between "fully marginal-cost based" and "fully price-history based" marginal cost definitions. A more accurate description of this split is between devices that do and do not have a state, either by energy storage, or from switching costs between operating modes. The control of a device with (relevant) state needs to reason about the cost over time. The "price-history based" bidding functions give heuristics for the cost, based on the execution history of the system and the agent developer's experience.

The central differentiating characteristic between single commodity Walrasian auction based smart grid control approaches is the method that is used to manage the storage within a cluster of agents, and the way that system objectives are embedded in the bidding functions. The outcome of the method determines the price set points of the devices, or the applied price transformations. For example, PowerMatcher uses local forecasts of the clearing price to determine “low” and “high” price levels [194, 195]. In recent work, this is refined with a sequential two time scale bidding process [182], and with a device level planning with stochastic optimization on the clearing price of the auction [265]. To embed objectives other than marginal cost optimization, e.g. to allow a VPP operation, a VPP agent (“trade dispatch objective software agent”) is used which attempts to express the objective of the VPP in terms of marginal cost [355]. In the Intelligator, the auction serves as a scalable dispatch mechanism. Similarly, a variant of TRIANA uses an auction as the mechanism for real-time control [232], and we discuss this variant in Section 2.4.2.2. In these “hybrid” approaches, where the control is split between a (cooperating) planner and a dispatcher, the control decisions impact each other [166]. Experience with these systems shows that the management of the available flexibility is crucial for the stable operation of these clusters, and that the controlled distribution of the use of flexibility over time is important for a stable and economic operation [74, 217, 218, 265, 355]. Note that this conclusion is similar for other types of control, e.g. based on planning, MPC, or stochastic optimization problems with recourse, which have to give up some immediate profit to avoid (possible) higher costs in the long run.

2.3.2.2. MULTI-AGENT SYSTEMS

A different perspective on smart grid control comes from the field of *multi-agent systems* (MAS) [367]. Where a top-down system optimization takes a perspective where “the world” needs a mechanism to make the DER work together, MAS aim to construct bottom-up control strategies with the same goal. An agent could be a physical person, but usually it is a system that acts on the behalf of a person or an organization. An agent can be as simple as a set-point thermostat which controls the heating system on behalf of the user to keep the temperature in a comfortable range, or as complex as a flash trading application which buys and sells stocks within milliseconds. The MAS paradigm applies to almost any field with distributed resources, wherein these resources are represented by an agent.

Agent based modelling gives an alternative conceptual framework to classify the control problem [277]. The distributed smart grid coordination problem fits well within this framework, in particular in an environment where “the operator” does not exist, or has very indirect control over resources. Whereas the conceptual framework differs, many of the concepts can be translated to an operational research or control framework, and solution techniques are shared between these domains (e.g. variants of DP with different symbols for the variables). Calling a system “agent based” suggests an awareness of the properties of the environment, a clear separation between different entities, and the need to study the emergent behaviour

of an agent over time and/or in a group of these agents at a macro level. In contrast, e.g. control theory relies more on mathematical proofs than on (simulation) experiments to obtain confidence in the suitability of a control policy. Agent systems usually aim for Pareto optimality, which means that no solution exists which all agents would prefer, i.e. decreases the utility (objective value) of none of the agents; clearly, this is often only a local optimum, unless the value is redistributed with suitable compensation mechanisms.

Agent based modelling characterizes task environments according to a *performance, environment, actuators, sensors* (PEAS) description [277: p. 38]. *Performance* describes the measure of success, or utility; this corresponds to the objective value and constraints in mathematics, and the ability to track a target in control. For example, in smart grids the performance metric may include reducing wear of the distribution grid, reducing energy use, and reducing BRP imbalance; as these are often conflicting objectives, it depends on who is involved in the operation of the system which ones are considered in the performance metric. *Environment* describes the (constraints on the) behaviour of the environment in which the system operates; this corresponds with the choice between a deterministic and a stochastic perspective (and the corresponding probability distributions) and the constraints in mathematics, and the modelling of disturbances in control. In smart grids, this would e.g. describe whether (involuntary) islanding needs to be considered due to limited reliability from the main grid. *Actuators* describe the abilities of agents to act on the environment; this corresponds to the variables and constraints in mathematics, and controllability in control. In smart grids, a battery is an actuator that can shift energy, but can (or rather should) not change the total energy content of the system. *Sensors* describe the options to observe the environment, which may vary widely in mechanism and in accuracy; in mathematics, this corresponds to the probability distributions on parameters, and open/closed loop systems in control. In smart grid, a controller may have real-time data on the consumption of each device, or only 10-second aggregate consumption data at a household level.

Smart grids take measures to improve the environment. For example, the availability of a trusted coordinator, i.e. an aggregator or the local DSO, adds structure to the environment. Energy cooperatives turn smart grids into a (locally) cooperative environment. DER add actuators that can improve the performance of the system. Sensors in heat buffers give an indication of their SoC. Creating an ideal environment is too expensive for large groups; consequently, distributed smart grids remain a challenging task environment.

Many practical environments are highly challenging, which may be addressed with control methods from *artificial intelligence* (AI) (or rather “computational rationality” [367: p. 17]). Challenging environments (or rather applications) addressed by AI include e.g. chess, number plate recognition, and automated driving. The goal of AI is to build intelligent agents that behave well in these environments. Game theory, which is often used to model interactions between players in a non-cooperative smart grid environment and in market-based power systems, is closely related to AI. In [272], the operation of smart grids is referred to as a “Grand Challenge” that should be solved by AI.

Agent-based models are pervasive in smart grids, e.g. [143, 262, 351, 370]. For agents in cooperative environments with a (de)centralized dispatcher or auctioneer, the distinction between an agent-based model and a “real” optimization is vague; consider e.g. [195, 350]. Whereas Molderink [231: p. 80] claims that TRIANA is not agent based, in particular the local control [231: p. 74] conceptually maps well to the agent framework. Game theoretic methods have received considerable attention as well, which lift the assumption that all agents have a common central objective; subsequently, the goal is to design mechanisms that give good outcomes in spite of this [112, 158, 219, 230, 282, 348].

Optimization, control, and AI strive for optimal control policies. In contrast, agent based simulation tries to characterize the behaviour of a control policy, regardless of how it is established (however, the goal of studying the behaviour is usually to improve it). Tools for agent based simulation differ in terms of which aspects of the system are taken into account. For example, EnergyPlus simulates energy streams in buildings in high detail [66], GridLAB-D simulates power systems [57], and Market Garden simulates the interaction of agents on electricity markets [60, 209]. To address the different aspects, a co-simulation approach may be used, i.e. link simulators that are specialized for different domains to build a coherent multi-domain simulator. Note that this approach is difficult in optimization, because the optimizer has to address multiple models concurrently. Furthermore, the models are in general too large and unstructured to address directly; appropriate abstractions enable the development of optimization algorithms that do have a reasonable efficiency.

2.4. TRIANA: DECENTRALIZED SMART GRID COORDINATION

TRIANA is a concept for decentralized coordination of distributed energy resources for smart grid applications, e.g. demand side management and virtual power plant management, which can deliver services such as peak shaving, market trading, or a combination of services. Based on this concept, algorithms and software implementations have been developed to deliver these services. Some of these implementations have been demonstrated in practical experiments [231: p. 132] and in field tests [20, 362]. Molderink [231] describes TRIANA as “the three-step control methodology for smart grids”. The TRIANA concept is very generic: different algorithms, resources, and applications have been demonstrated in simulations.

The central idea of TRIANA is scalable, hierarchical planning. At each hierarchical level (e.g. transformer, house, device), the planning makes an explicit representation of the expected energy supply and demand profile over time, and applies steering signals to improve the shape of the expected profile. We show this structure in Figure 2.3. The planning follows an iterative process, i.e. with multiple planning rounds: the outcomes of the previous rounds are used to refine the steering signals. We believe that planning is essential for an economical operation of energy systems to avoid short sighted behaviour. The planning works on a certain horizon; for VPP market integration, this is usually a day-ahead horizon.

FIGURE 2.3: Partitioned planning approach in TRIANA.

The planning of TRIANA relies on a prediction of the energy exchange profile under the given steering signals. The prediction follows a bottom-up approach, with aggregation at each hierarchical level. The bottom (i.e. device) level predictions rely on a model of the available energy flexibility of the device at hand. The models used for prediction are fairly explicit and device class specific, similar to the plant models in the unit commitment and economic dispatch problems. By this, the model can account for device specific constraints (e.g. minimum run times, maximum number of starts, ramping constraints), and can avoid undesirable interactions between devices.

The device models rely on forecasts of their parameters. For example, to predict the electricity production of a microCHP with a heat buffer under some steering signals, TRIANA makes a planning of the runs of the device over time. This planning uses an estimate of the parameters of the device, e.g. the start-up behaviour, the electricity output, and heat production; we assume that these parameters are (almost) static and thereby not considered part of the forecast. The most dynamic parameter in the model that is subject to forecasting is the heat demand [19: p. 37]. Similarly, EV planning uses a forecast (or statement) of the arrival time, charging deadline and the minimum charge amount for the vehicle. For devices that TRIANA can not control, e.g. the demand of smaller household devices and the production of RES, the forecast determines the prediction without planning (some RES support curtailment, which is controllable and can therefore be planned).

In principle, the planning gives a policy to control the devices, e.g. a time schedule of the device mode switches, or the values of control set points over time. Every device has its own policy, and can execute their plan without further communication. However, this approach does not consider dynamic interactions between devices, which are particularly relevant at a local level: for example, to exploit energy locality, an energy consuming device should run when the sun shines, instead of when the sun tends to shine on average. Similarly, correlated events, e.g. a cloud passing over a neighbourhood with a large penetration of PV, can not rely on statistical averaging; these events need to be addressed dynamically. This dynamic response is called *real time control*, or *operational control*. The operational control replaces the policy of the planning with a dynamic dispatch policy that aims to

minimize the distance to the predicted profile at each hierarchical level, considering only the instantaneous conditions. While this dynamic, greedy approach improves balance in the short run, it risks to (and often does) increase imbalance in the long run, because it may exploit flexibility too early (e.g. similar to the local PV storage in Appendix A.2.2.3.6 and the auction control in Section 2.3.2.1.5). Furthermore, the dispatch may give unintended behaviour and interactions between devices. For example, it may respond to a short period of sunshine by starting a group of washing machines, which are then committed for the coming hours. A short-term household level MPC is proposed in [231: p. 78] to consider these interactions, although at an excessive computational cost). Whereas it is possible to guard against poor control actions with large safety margins (e.g. 10 – 20% of buffer capacity at both ends as in [19: p. 121] – which costs 20 – 40% of the capacity; a factor 3 overdimensioning may remove the imbalance almost certainly without control [40: p. 119]), this is economically unattractive. Moreover, the problem is often not the amount of demand, but its timing: addressing a “horizontal” (i.e. time) problem with a “vertical” (i.e. SoC) margin may not be the most effective approach, comparable to the way that the conventional control policy for PV storage needs vast amounts of storage to reduce peak demand (see Appendix A.2.2.3.6).

The system can detect excessive excursions from the planned profile (e.g. 10% measured absolute difference [19: p. 82]), and can perform “re-planning” [231: p. 83] as a response to these deviations (this is analogous to redispatching in the power systems literature; see Section 2.3.1.1). The method gives a variant of rolling horizon optimization that is modified to limit the number of replanning operations; [19, 231] consider planning as an expensive operation, which should be limited to e.g. 10 sessions per day at most. Consequently, the replanning responds only after the flexibility has already (locally) been spent. In spite of this, replanning gives an improvement in objective performance [19: p. 133].

2.4.1. PLANNING

The planning of TRIANA uses different planning methods at different hierarchical levels. We describe the “typical” static multi-level configuration that is described in [19: p. 63, 231: p. 70, 40: p. 75]. We illustrate this structure in Figure 2.4, and explain the elements in the following.

The central coordination mechanism is *iterative distributed dynamic programming* (IDDP). IDDP implements a variant of Lagrangian relaxation (introduced in Section 2.3.1.3.1), with several practical modifications. Alternatively, it may be considered as a combinatorial auction that only supports block bids. In terms of design, IDDP follows the VPP paradigm, with some changes to support a DSM context. A comparable idea is presented by [2] in a more formal context. The master problem splits up its optimization time horizon into time intervals (e.g. 96 intervals of 15 minutes for a 24 hour horizon), and maintains the sum of expected supply and demand of all subproblems within each time interval. The environment can set an upper and lower bound on the outcome of the sum for each time interval; if no bounds are provided, then the algorithm chooses a flat reference profile with the bounds at a small distance from this profile (e.g. 10%), based on the average

FIGURE 2.4: Realization of planning hierarchy in TRIANA corresponding to Figure 2.3.

consumption in the first planning iteration. The microCHP VPP optimization case in [40: p. 75] optimizes for a given price profile, and subsequently tries to find a feasible solution within an upper and lower bound at low extra cost (i.e. with only small price changes), with a problem-specific nodal price update rule [40: p. 80]. In [19: p. 74], the focus shifts to profile shaping: the bound violations are interpreted as the actual costs, and the electricity prices are used as a starting point of the search (usually, a flat price profile is used). As a generic price update rule, [19: p. 74] introduces randomized nodal pricing to avoid a synchronized price response from “uniform pricing” (note that time intervals do have different prices), and to avoid overshoot from a concurrent price update. Similarly, [231: p. 72] considers minimizing the quadratic distance from a reference profile as the objective, which is approximated with an upper and lower bound profile at a fixed relative distance from the reference profile. The algorithm converges typically (but not always) in 10 – 20 iterations [19: p. 80]. The objective value corresponding with the resulting profile does not decrease monotonically; IDDP retains the solution with the lowest cost. In the applications that focus on profile shaping, the nodal prices should be interpreted as steering signals or virtual prices, rather than as a dynamic tariff that is actually paid by the user; there is no direct correspondence between the “true” value of the profile and the stated value within the subproblem (to avoid excessive price differences, the difference between the market price and the nodal price may be bounded).

IDDP is a simple, somewhat flexible, and efficient mechanism. By setting appropriate bounds and prices, the mechanism may for example be used for peak shaving (positive and/or negative), following a profile, self-consumption of local generation, real time pricing optimization, and (some) combinations of these objectives. Hoogsteen [159] adapts the update rule of IDDP to account for voltage constraints in the distribution grid, using load flow analysis. However, the mechanism does not always work particularly well for real problems: the objective of the optimization mechanism differs from the objective of the user. Bosman proposes an ADP approach [40: p. 85, 341] (which is comparable in solution strategy to a lumped modelling approach; see Section 2.3.2.1.4) and a column generation approach [40: pp. 91, 148] as alternatives, both of which offer promising results.

IDDP is typically used at a single hierarchical level, e.g. the distribution transformer level, by what is referred to as a *grid controller* (note that a nested operation of IDDP is feasible, although not implemented). To adapt multiple devices in a house to a single steering signal, the *house controller* distributes the prices to all devices, and sums up their electricity profile predictions. The lack of balancing and the single price profile at a house level reinforces local supply and demand peaks, which *introduces* local voltage problems, and is thereby not suitable for DSM [160, 198, ^{HT}6] (whereas these local issues are transparent from a VPP perspective). Above the transformer level, Bakker [19: p. 63] proposes to support a hierarchical operation by dividing the contribution to the upper and lower bound proportionally over a group of IDDP instances: if every IDDP instance manages to meet the bounds, then the global bounds are met as well. The controller at the top of the hierarchy is called the top grid controller, the IDDP instances are called bottom grid controllers, and there may be middle grid controllers in between. The main motivation for this approach is scalability: each IDDP instance can plan a group of devices independently. The approach assumes that the bounds are permissive enough to match with the subgroups, and that the subgroups have a similar population to allow for an appropriate distribution of the bounds. In [160] a similar approach is used for power quality management. This paper clusters houses on the same feeder and AC phase to avoid phase load imbalance in the distribution grid: a separate balancing of each phase is less susceptible to imbalance than a balancing that is unaware of the grid structure. For example, if some specific phase is highly loaded, then a grid unaware balancing approach may try to compensate for the high aggregate load by decreasing the load on a different phase, which further increases the phase imbalance.

The bottom, device level planning problems (i.e. minimize the sum of electricity cost and local cost, subject to nodal prices) are solved by DP. The planning problems of devices are considered as completely separate optimization problems; if multiple devices are coupled from a control perspective (e.g. a microCHP, a heat buffer, and a heating rod), then these are represented as a single device in the planning. For uncontrollable devices, the “planning” corresponds to its supply or demand forecast.

The demand prediction of the controllable devices follows from the solution of the planning problem, by “simulating” the behaviour of the plan, using the same scenario as the planning problem. Usually, the simulation is a trivial operation, e.g. copying an expected demand vector. The planning and the subsequent demand prediction choose the most probable scenario, i.e. with the highest expected value for its parameters (assuming that the forecast works as intended). In the light of the discussion on “internal” uncertainty of the DR resources from Appendix A.2.2.3.3, a less optimistic prediction would be more appropriate: predicting that every device will work exactly as planned is a naive representation of the expected value of the load distribution over time, which should consider the behaviour of the control policy under more than a single realization.

FIGURE 2.5: Example of house cost control energy stream model [HAT:1].

2.4.2. OPERATIONAL CONTROL

Two variants of operational control are commonly used with TRIANA: a mechanism that we describe as house cost control [231: p. 37] (the original work describes this method as ILP control in accordance to its solution method, but we prefer to use a different term to avoid confusion with the unit commitment-like MIP methods), and a Walrasian auction based dispatch mechanism [232]. Both methods will be discussed shortly in the following.

2.4.2.1. HOUSE COST CONTROL

House cost control describes the available control flexibility within a household. A multicommodity energy stream model describes the configuration of connections between devices, and the available control options for devices. This model gives a bipartite graph of devices (energy producers and consumers) and pools (balance equations), connected by streams. Implicitly, devices have ports, e.g. separate heat ports for the heat production and thermal losses of a microCHP. Pools, streams, and device ports that are connected need to be of the same type (i.e. have the same commodity). An example of this model for a heat pump system is presented in Figure 2.5. A more extensive model is given in Figure B.1a. A device presents a mutually exclusive set of control options. A control option gives an operation range with a fixed energy stream ratio between ports (e.g. a microCHP may offer an option with a control range 500 – 1000 (unitless), in a ratio of 10 W : 8 W : 1 W on the gas, heat, and electricity ports, respectively), a fixed cost for choosing the option, and a marginal cost per range unit (note that a device only considers the cost for the device itself, e.g. due to wear and subjective user cost, and not the cost for the commodities). Some devices offer multiple of these control ranges, e.g. to allow for a separate (variable ratio) decisions on the input and output of a heat buffer. The control options of the devices have to be cleared simultaneously, such that the balance equations of the pools are satisfied. This is in contrast to the planning model, which treats the control options of devices separately, and considers only the aggregate electricity consumption.

Each device has a bidding function generation controller (i.e. an agent) which generates the control options: for example, a microCHP prefers to run when gas is cheap and electricity is expensive, and a heat buffer “prefers” to be discharged when it is almost full and charged when it is almost empty (note that the microCHP agent and the buffer agent are separate in this model). This agent model is highly comparable to e.g. PowerMatcher (compare [231: pp. 87–88] and [195: p. 169]), but is more expressive. An agent should try to follow the planning, which is represented as a “planner preference” (e.g. the planned electric demand of a device); usually, this preference is expressed in the control model as an extra single-point control option with a negative cost, such that it is always chosen unless the system constraints do not allow this.

At the periphery of the house model, exchanging devices control the exchange of energy with the grid (there are three other device categories – converting, buffering, and consuming devices – which serve to coalesce tightly coupled devices in the planning stage). The cost model of the exchanging devices determines the cost for the import or export of a certain commodity. The exchanger specifies a linear cost for the import and export of commodities; the commodity cost can be changed by the house controller and the grid controller to perform open-loop control (i.e. a higher price can decrease demand, but the magnitude of decrease in demand is not known in advance). The exchanger cost model does not directly consider the local house-level demand prediction, but rather relies on the behaviour of individual devices to follow their individual predicted profiles. The bidding strategy of the agents, the grid controller, and the house controller need to be tuned to each other. To facilitate this tuning, a normalized commodity price level (i.e. “1000 units”) is used, which device agents can use to reason about “high” and “low” prices. The model in principle sequentially considers a single decision point in time; the tuning mainly serves to reach the desired interaction between devices over time. An MPC extension is available to reason ahead a few time intervals [231: pp. 78, 128], yet requires a manual specification of the control options over time by the agent, and is therefore rarely implemented. Furthermore, the mechanism usually gives a linear or quadratic increase in (binary) variables with the horizon length, and is thereby computationally expensive; a DP-like memoization structure is used to avoid an exponential growth in variables (as illustrated in [231: p. 80, Figure 4.5]: the dependency graph is a DAG with shared nodes rather than a tree).

The house cost control can introduce a nontrivial optimization problem. To allow mutual exclusion between the piecewise linear control ranges, the model uses binary variables;⁵ the problem size depends on the number of devices, and the structure of the model that the agents of these devices generate. The house controller formulates the problem as a MIP, which may have an unpredictable optimization execution time (i.e. milliseconds or minutes for seemingly similar problems), in particular when there are many binary variables.

5. Optimization problems with binaries, or more generally integer variables, are often much harder to solve than linear problems: the solution time grows exponentially in the number of integer variables in the worst case.

The house cost control mechanism allows for a dynamic reallocation of flexibility within a house, and for a coarse control of the demand of households with a marginal price signal. The control can balance between following the planning and following immediate disturbances, depending on the device agent bidding strategy. By this, the method can be fairly robust against local disturbances. However, developing bidding strategies for device agents that give appropriate behaviour appears difficult in practice.⁶ Also, this agent-based approach virtually excludes (mathematical) reasoning about the behaviour, as there is no explicit model of the response with convenient structural properties (or the agent does not expose this model). The profile steering by the grid control and house control mechanism (using an exchanger device agent) have been exploited only partially, i.e. these do not directly attempt to match the predicted profile (as a benefit, this reduces the risk of over-using flexibility as in Section 2.3.2.1.5). Due to the separation between the planning and the control phase, the prediction (which is based on the planning) gives limited insight in the expected behaviour of devices. The solution method for the real-time control problem (mixed integer (linear) programming) gives unpredictable execution times in both the worst case and the average case, which is a well-known limitation of MIPs [167]; other solution methods that give approximate results or exploit more problem structure may be more suitable to ensure a real-time operation.

2.4.2.2. AUCTION CONTROL

As already mentioned in Section 2.3.2.1.5, TRIANA also offers a different operational control mechanism, based on the Walrasian auction. Just as the planning model, this control method only models aggregate electricity demand, rather than the specific energy streams within the house (the house cost control method does model energy streams within the house). Analogous to the house cost control, the auction control can incorporate a planning (i.e. follow the prediction of TRIANA) with bidding curve transformations (e.g. inserting a “bend” at the desired consumption level in the bidding curve) at multiple hierarchical levels, and may also incorporate a price sensitivity based droop control to adjust supply and demand dynamically. In contrast to the house cost control, the auction mechanism scales naturally beyond house control.

The auction control offers a dramatic improvement in peak shaving performance over house cost control [232: Figure 6]: the controller is able to actively exploit the diversity between devices in a scalable, efficient way. However, it inherits the drawbacks from this auction mechanism (Section 2.3.2.1.5), i.e. market restrictions (single commodity, nondecreasing/nonincreasing bids, only marginal cost), difficulties to express objectives, and the need for an always available communication (as a fallback, a local market may be started; a local planning may be used as a reference profile). Furthermore, it shares many shortcomings with the house cost

6. The experimental method to find the parameters that work for the case at hand is colloquially referred to as *black magic*; note that the parameter search may be replaced by a learning process, e.g. as in [62].

control method, e.g. shortsighted decision making, difficulty to formally reason about the response, and objective specific bidding strategy development for each device. The latter has led to the development of *generic* (class specific) agents in the context of PowerMatcher. In Chapter 5, we show that the underlying framework is flexible enough to support TRIANA as well, with a different type of generic agents.

Similar to the house cost control, the auction can respond to disturbances, and the magnitude of the response is controllable. Depending on the sensitivity of the cost functions, the method can trade off between adhering to the planned consumption pattern in the short term, and decreasing the (perceived) likelihood of failure in the long term by adhering to the planned (or a neutral e.g. half-full, or biased e.g. near-full) state. Nutma [246] presents a method where the auction is used to follow both the planned global demand profile and the planned local demand profile of devices. However, it is not clear which strategy (e.g. droop weights and set points) should be used. Note that “the best” behaviour depends on the true scenario at hand, which is not known on beforehand; an appropriate policy should address all reasonable scenarios.

2.4.3. EVALUATION

The review in Section 2.4.1 and Section 2.4.2 shows that TRIANA is a promising approach for decentralized smart grid control. However, the “three-step control methodology” – forecasting, planning, and operational control – may benefit from more integration between the steps, in particular in environments where uncertainty plays a substantial role. The forecasting gives different information than the planning would most benefit from: for example, the planner would prefer to have a probability distribution of expected arrival times of an EV, rather than the expected value of the arrival time; if the planner tries to plan it as early as possible according to this value, the plan is infeasible roughly half of the time (depending on the shape of the probability distribution). The local, planning based prediction gives an incomplete characterization of the expected supply, demand, and operational flexibility of individual devices, which adversely affects the ability of the coordinator to evaluate combinations of plans. Conversely, the coordinator has a limited ability to communicate its intents to the subproblems: even though the desired local response may be obvious from a central perspective, the mechanism may be unable to express it [198, ¹⁴T:6]. A concrete example is the battery storage case in [198: p. 49], where linear pricing induces large supply and demand peaks: except for time intervals where the battery is near full or near empty, there is no incentive to charge less than the full rated power (as a workaround, this case uses a quadratic regularization, or “politeness” term in the local cost function to avoid excessive behaviour). More formally, if we consider the local planning problem as a function of prices, then the range does not include the desired response; with modifications of the cost function of the local planning problems, the range can be changed to include more appropriate response values (within the constraints of the local problems).

The planning step and the control step have a different model of a building: whereas the control model views a building as a multicommodity energy stream graph, the planning model considers only hierarchical electricity streams. This disconnect limits the ability to use the expressiveness of the control model, while it still incurs the cost of the control model. Furthermore, the dynamic policy of the control step offers limited insight in the consequences of control actions in relation to the planning. Hereby, the operational control makes the demand behaviour of a building, and eventually of a cluster, less predictable.

The planning and control of TRIANA rely on the idea that prediction errors cancel out at a cluster level (provided that the demand prediction indeed represents a prediction of demand, rather than a single scenario). Central dispatch based methods (including most aggregate models from Section 2.3.2.1.4 take a more proactive approach to manage the behaviour at a cluster level. Therefore, we should address the ability to redistribute flexibility spatially, i.e. between different devices in a cluster. However, this is not always possible, in particular in islanding scenarios, cases with communication problems, and for correlated errors (e.g. with an unanticipated overcast sky, all houses in a neighbourhood have a lower than expected PV production). Therefore, a temporal redistribution of flexibility is important as well. The power systems domain (Section 2.3.1.3.3) and the smart grid coordination domain (Section 2.3.2) provide interesting options for addressing uncertainty.

We first address the hierarchical structure and the conceptual interfacing between components (houses, devices, transformers) and between optimization models in Chapter 3, which gives a unified model for planning and control.

2.5. CONCLUSION

The energy transition has a dramatic impact on almost all aspects of energy: production, consumption, transportation, and distribution. Energy moves to the electric domain, which increases the load on the electric infrastructure; furthermore, renewable energy sources give a variable production of energy (Appendix A.1).

These changes motivate smart energy systems and smart grids, which can exploit the differences between energy types, improve the utilization of the infrastructure, and adapt the demand to the available supply over time. Demand side management is an important source of flexibility for the smart grid (Appendix A.2).

Smart energy systems and smart grids rely on smart control. The coordination of energy resources is a grand challenge, in particular for large scale demand side management applications. The conventional (inter)national electric power systems pose similar challenges, and have been well studied in literature. Smart grids have different constraints and may therefore be operated in different, more dynamic ways that better match with the variability of renewable energy sources and with locality. A major problem in decentralized smart grid control is the limited predictability of the control, which leads to reliability issues (Section 2.3).

TRIANA is a decentralized demand side management/virtual power plant control approach for the economic operation of distributed energy resources. The approach strives for predictability by making an explicit schedule of the use of devices

over time. Subsequently, a dispatcher takes online, operational control decisions to address the variability of uncontrollable supply and demand, and to cope with wrong estimations of the available flexibility in controllable devices. The disconnect between the planning and the dispatch can lead to unpredictable behaviour in practice. Therefore, the planning should account for the dispatch mechanism; consequently, the dispatch mechanism should be chosen such that the planning can account for it. We still believe that planning is a suitable mechanism to control and estimate energy streams over time; consequently, we propose improvements to TRIANA in the rest of this thesis (Section 2.4).

In the following chapter, we propose a structured model for hierarchical, planning based demand side management for use with TRIANA, which enables hierarchical composability. We apply this model in Chapter 3 and in Chapter 5. $\mathfrak{S}$

COORDINATION IN TIME

Abstract

This chapter treats demand side management as a large scale variant of a unit commitment and economic dispatch problem. We discuss three optimization based planning heuristics with different properties, and develop a common hierarchical model for these methods. IDDP is a highly scalable heuristic, column generation is a very flexible approach, and profile steering leads to very smooth demand profiles. We compare these methods in the Flex Street simulation case, which models 400 households for the duration of a year. The planning methods may give near-optimal results in large scale cases (e.g. at neighbourhood level), and only profile steering gives good results in smaller scale balancing cases (e.g. at household level). The results of this chapter are applied in Chapter 5.

model 45 · IDDP 66 · column generation 74 · profile steering 107 · experiments 117

• • • • •

3.1. INTRODUCTION

COORDINATION poses one of the main challenges for DSM within smart grids. As a first requirement for DSM, devices have to expose their flexibility, and once this flexibility is available, it has to be controlled. Although it is in principle possible to control devices independently of each other, a coordinated control approach may help to make the best use of the available resources. For example, if we have PV feed-in, an EV that needs to be charged, and a battery, then in most cases the PV production should be used to charge the EV first, and then the battery. However, matters become less simple if we think beyond the boundary of a single house. Before charging the battery, we may consider to sell excess production to help charging the EVs of the neighbours as well, but only if they are willing to pay enough

↪ Parts of this chapter have been published in:

- [^HT:3] “Column generation based planning in smart grids using TRIANA”. In *ISGT Europe*. IEEE PES, Oct 2013. ISBN 978-1-4799-2984-9. DOI 10.1109/ISGTEurope.2013.6695431.
- [^HT:4] “Cascaded column generation for scalable predictive demand side management”. In *EnergyCon*. IEEE, May 2014, pp. 1228–1235. ISBN 978-1-4799-2449-3. DOI 10.1109/ENERGYCON.2014.6850580.
- [^HT:6] GERARDS Marco ET, TOERSCHE HA, HOOGSTEEN G, van der KLAUW T, HURINK JL & SMIT GJM. “Demand side management using profile steering”. In *PowerTech*. IEEE PES, Jun 2015. DOI 10.1109/PTC.2015.7232328.

and the grid is not too congested. Conversely, the neighbours may speculate on whether we have excess supply to sell in the future and decide to postpone their demand, which may lead to a demand peak in the future if all neighbours make the same decision. With planning (i.e. coordination over time), we can prevent many of these incidents, and thereby reduce cost.

The coordination of resources (devices) in a neighbourhood poses a significant challenge, because there are many possible dependencies over time and between devices. In fact, the coordination scheduling problem is already NP-complete for a limited case [40: pp. 52–57], which indicates that we should look for approximate methods to address this problem. In this chapter, we focus on the efficient coordination of resources regardless of who owns them, and leave the market mechanism (which arranges the fair wealth distribution) as a topic for future work.

We start with the observation that the coordination problem within DSM has many similarities to the problem of the coordination of power plants. This leads to the idea that we may schedule the devices as a virtual power plant using techniques similar to those with which real power plants are scheduled. This chapter follows the line of the work in Chapter 3 and 6 from the PhD thesis of Bosman [40], which is related to the conventional scheduling methods within the grid at transmission level, addressing unit commitment, economic dispatch and optimal power flow problems, as discussed in Section 2.3.1. A core principle in these methods is *decoupling*: instead of scheduling the units all at once (which leads to a huge optimization problem), units are first scheduled individually over time, and then receive feedback on what the other units propose to do. The planner can determine a coherent, coordinated schedule for the units by performing this procedure iteratively. These scheduling problems posed mainframe scale computational challenges in the early 1980s [228], but can nowadays be easily solved on desktop class hardware. Since then, more coupled scheduling methods (which consider the temporal and spatial relations between units at the same time) have become commonplace in the transmission world as orders of magnitude more processing power and algorithmic improvements have become available, allowing for percentage-wise small cost reductions that nevertheless translate to tens of millions of euros in savings annually at national scale.

However, there are some important differences between DSM coordination and power plant coordination. One of the main differences is that for a DSM problem there are in the order of 1000000 times more “units” to consider, each of which is in the order of 100000 times smaller than a conventional power plant in terms of power exchange. This makes it evident that the computational budget to optimize each resource should be relatively limited. Also, it should be clear that (directly and individually) optimizing millions of devices at once is infeasible or at least very expensive, let alone to communicate the decisions to each of these devices. Furthermore, as a central scheduling approach gives significant reliability and privacy concerns, we should clearly aim for a *decentralized* scheduling approach, and decouple the optimization problem to support a decentralized implementation of the optimization approach. This motivates the development of a spatially decoupled scheduling algorithm for DSM.

FIGURE 3.1: Basic overview of decoupled coordination.

Another important characteristic of DSM is that the energy volume is generally more or less fixed: where power plants can choose *how much* to produce within some constraints, DSM resources only have the freedom to choose *when* to consume within a certain time period. The energy volume can change if there are losses, local generation or hybrid/multicommodity devices, although these options may lead to higher cost. This limited and sometimes expensive freedom makes the DSM coordination problem more challenging.

To give a basic first idea, consider the example in Figure 3.1, where we have to schedule a group of three controllable devices.¹ The goal of the optimization in this example is to reach a demand profile that is as flat as possible. Formally, this goal can be reached by considering the minimum and maximum demand values, or the average quadratic distance to a demand of 0. Device 1 and 2 both give an uninterruptible demand of 2 kW for a duration of 6 h, each of which may represent a (very slow and inefficient) washing machine. Furthermore, device 1 must be started between 6:00 and 15:00, and device 2 must be started between 9:00 and 15:00. Device 3 has a total electricity demand of 9 kWh, and may consume between 0 and 2 kW from 3:00 to 21:00. This device may represent e.g. the charging of an EV. Initially, no information on the interaction between the sched-

1. The example closely follows the approach of Section 3.5, and we consider the same example more formally in that section.

ules of the devices is given. As a first attempt, we may let device 1 and 2 start as soon as possible, and let device 3 spread out the load over the available time period. The corresponding demand profiles over time are shown for each device in the lower plots of Figure 3.1 (marked *o*). When we sum up these profiles, we get the aggregate demand profile in the upper plot (*o*), which has a large demand peak between 9:00 and 12:00. We now can ask each of the devices to propose an improvement, based on the estimated aggregate demand profile. Device 1 proposes to start at 15:00 (the latest possible time), such that it no longer overlaps with device 2. Device 2 should propose to start somewhere between 12:00 and 15:00, such that it no longer overlaps with device 1; we choose to start at 12:00. Note that while the figure seems to suggest that device 2 should start at 15:00, we should first subtract the demand that device 2 has contributed to the aggregate demand profile, which lets the period of low demand start after 12:00. Device 3 should evenly fill the “gap” around the peak of device 1 and 2, meaning that it no longer contributes to their peak (visually, the demand of device 3 behaves as a body of water in a container from 3:00 to 21:00 with an “iceberg” of other demand in the middle). We mark the proposed profile in both the upper and lower plot by their device number (1, 2, and 3). The changes proposed by device 1 and 2 both result in the same large peak reduction. The change proposed by device 3 gives only a small improvement, as it contributes only 500 W to the peak. If we now decide to accept either the change of device 1 or 2, and then plan device 3 around this pattern, then we find an optimal profile (not illustrated).

Note that while in this example we have a complete overview and direct control over the devices, in practice the group of devices is generally too large to “micro-manage” every device. Therefore, devices are controlled indirectly with steering signals, for example with prices.

Another important aspect is the controlled application of the changes proposed by the devices. If we aim to apply the changes of all devices at the same time, then this may make the peak demand even larger (5 kW between 15:00 and 18:00, not illustrated). This is a prevalent problem if we control devices using a common price profile: if we tell that electricity is cheap between 15:00 and 24:00, and we have a lot of flexibility, then we may introduce a new demand peak in that period. Therefore, the planning approach proposed in this chapter “measures” (through simulation) what happens if we apply a certain steering signal, adapts the signals accordingly, and may choose to use different steering signals for different devices.

The work in [40] proposes two spatially decoupled coordination methods for DSM: a DP based local search method [40: pp. 78–85] also known as IDDP [233: p. 6], and a *column generation* based method [40: pp. 91–112, 141–178]. In their words, these methods exploit a separation of dimensions in time and in space (devices): the planner first determines feasible decisions over time at a device level, then considers how these decisions affect the cooperational constraints (the balance of electricity supply and demand) at an aggregate “fleet” level, and then determines how the device level decisions should change to satisfy the cooperational constraints. This corresponds with what we call spatially decoupled coordination.

In this chapter, we further develop the column generation method, as [40] indicates that this method gives the most promising results for implementation in a real-world setting. We place this method in a common framework, along with the IDDP method from [40] and the *profile steering* method from [HT:6], and extend this framework in several directions to meet new challenges (multicommodity planning, nested configurations, internal costs, nodes with few children), which are mainly motivated by the practical problems mentioned in Chapter 5.

At the end of the chapter, in Section 3.6, we evaluate and compare each of the presented optimization methods in a case study with the Flex Street case (Moderate variant), where we simulate the planning and control of a variety of flexible devices in 400 households. This case is described in more detail in Appendix B. The objective of the control for this case is to minimize the peak demand and to flatten the demand profile. In earlier work, we performed a simulation study where we compare (legacy versions of) TRIANA (using IDDP) and Intelligator [HT:8]. The results of this study are also summarized in the appendix. This comparison shows that the uncoordinated approach gives very poor performance, and IDDP and the auction-based coordination both have a comparable performance.

In Appendix B.3, we provide a model for the scenario, and derive lower bounds for this scenario in Appendix B.6, which we use for the evaluation of the case in Section 3.6. In addition to the year simulation case, we also use the case for some smaller experiments within this chapter. We also use the case for the experiments that will follow in Chapter 4.

This chapter is structured as follows. First, we present a general model for spatially decoupled coordination in Section 3.2. We shortly discuss IDDP (Section 3.3), and justify the development of improved optimization methods. Subsequently, we develop a column generation approach to solve the coordination problem in Section 3.4. After this, we introduce the profile steering approach (Section 3.5). We compare these three methods in Section 3.6, and also compare the results to the lower bound that we determine in Appendix B.6. We end with conclusions and recommendations for future work in Section 3.7.

3.2. HIERARCHICAL MODEL

3.2.1. INTRODUCTION

Before we consider the algorithms for spatially decoupled coordination, we first introduce a model that is used as a base for such algorithms. The earlier work on TRIANA already introduced related models for energy streams. The model presented in this chapter builds on these models, but aims to overcome their drawbacks. One possible candidate model would be the multicommodity energy stream model of Section 2.4.2.1, which is very expressive, but it is unclear how this model can be incorporated efficiently and effectively in a planning procedure. On the other hand, the planning models of [40] are tailored for the considered case and the solution method at hand, and it may therefore be not that straightforward to apply them more generally.

We believe that the planning somehow has to take into account the way that the control is organized. By this, these two steps should use an (at least partly)

common model, or have a well defined interface: as discussed in Section 2.4.3, the “forced marriage” between the control model and the planning model may lead to incoherent behaviour if the planning model does not describe what happens in the control phase. We aim for a model with an expressive power comparable to the first model, and the “planning friendliness” of the latter. Instead of trying to come up with an overall model that encompasses everything, the presented model should be viewed more as a framework to set up hierarchical planning and control schemes. Note that even though the model is general, specific solution methods may impose extra restrictions on (part of) the structure of the model. We furthermore suggest to use the optimization model in a rolling horizon approach, where we periodically perform a (re)planning session over a restricted time horizon. Although there are opportunities to reuse information between overlapping parts of planning sessions, we have not yet exploited this.

In the development of the model, we aim for a decentralized structure. This structure allows the optimization problem to be split up over multiple embedded computers that can be located close to the considered resource (e.g. a household or substation), and may allow for local fault tolerance e.g. by trying to continue to realize a certain “promised” demand profile if part of the communication infrastructure is unavailable. This splitting implies that communication between the different computers of the system is needed, which can have a relevant cost if we communicate too much. Note that the communication standards that are currently considered for smart grid applications (e.g. GPRS) are orders of magnitude slower and more costly (per MB) than typical consumer Internet connections. We consider communication costs only in a qualitative manner (Bakker [19: p. 81] considers these costs quantitatively for the IDDP method that we discuss in Section 3.3).

Next to these more structural properties, we aim for a model where the decisions are in principle not affected by the choice of units or the choice of the time interval length (up to accuracy differences). If we only have electricity costs, then we may sometimes choose to optimize with arbitrary units, as long as it is used consistently. For example, if we have a fixed time interval length $\tau = 900$ s (which we also consider as a “unit” of time), prices c in €/kWh and a demand x in W ,² then $c^T x$ implicitly gives the cost in $W\tau$ €/kWh = τ €/kh (over a period of length $|x| \cdot \tau$), a cost unit that is clearly awkward and error-prone to work with (to translate this cost to €, we should multiply it by $\frac{900}{1000 \cdot 60 \cdot 60}$).³ A unit-correct model allows to give a direct relation between the objective value of the problem (which is in principle unitless) and the actual expected cost over the horizon (in €), up to where the model is accurate. This accounting problem becomes more prevalent if we want to reliably consider multi-objective problems, which consider e.g. the start-up costs of devices, the use of multiple commodities, and quadratic terms in the objective (e.g. to model electric transport losses or to penalize control deviations). The proper account-

2. We express vectors with bold symbols, e.g. $x = (x(1), \dots, x(n))$ with $n = |x|$. Furthermore, we often write $x + y$ for an addition of vectors $(x(1) + y(1), \dots, x(n) + y(n))$, with $n = |x| = |y|$, and $x^T y$ for $(x(1) \cdot y(1), \dots, x(n) \cdot y(n))$.

3. Currencies may be substituted with a scalar multiplication by the exchange rate, e.g. $(\$/\epsilon) \cdot \epsilon/\text{kWh} = \$/\text{kWh}$.

ing for a variable time interval length also becomes feasible when we account for units properly (and apply in Chapter 5). We can avoid many of these error-prone problems if we first go back to basic units (e.g. $\text{€}/\text{Ws} = \text{€}/\text{J}$). Note however that for numerical reasons, it may be necessary to prescale the values to an acceptable range. For manual inspection, it may be convenient to choose a familiar scaling factor, e.g. such that prices are represented in $\text{€}/\text{kWh}$ (this seems to contradict with what we stated earlier, but now we have a scaling factor that is introduced deliberately).

The remainder of this section is structured as follows. Before we go into the definition of the model, we first give an introduction at a conceptual level in Section 3.2.2. After this, we discuss the model more formally in Section 3.2.3.

3.2.2. OVERVIEW

3.2.2.1. BOTTOM-UP DEMAND PROFILE

We develop a model that describes the decentralized planning process of TRIANA in a general way. We model the electricity infrastructure with a tree that mirrors the hierarchical structure of the electricity infrastructure. The basic building block of our model is a node. A node in the tree may for example represent (the substation of) a neighbourhood, a household, or a device. Typically the leaves of the tree represent devices and generators, and the internal nodes represent groups with their associated infrastructure, e.g. transformers and cables. Nodes do not always directly correspond to physical infrastructure elements: for example, in a VPP context it is common to look at the aggregate electricity demand of a large dispersed group of devices at a transmission level, which is described by a “grid” node. If the decisions of a group of devices are tightly coupled, then we choose to add a node that combines these devices: for example, to schedule the operation of a microCHP, the operation of the connected heat buffer should also be considered (depending on the optimization method used, it may be necessary to develop an integrated optimization method for the combination as well).

We model the electricity demand of each node as a profile of average power consumption values over time on a certain horizon. A node contributes its electricity demand to the demand of its parent, which in turn contributes the demand to the parent above it, until the root node determines the external aggregate demand of the entire group. Nodes may also supply electricity, which is expressed as negative demand (and cancels out normal demand). We generally assume that it is always possible to push demand upwards; in a closed (e.g. islanded) electricity system, the aggregate external demand must be zero at all times (we will not address this case, as this requires hard real-time control). In principle we assume a radial distribution grid, although some non-radial (network) grid configurations may be supported as well by embedding the network in a single node.

3.2.2.2. PRICE/STEERING SIGNALS

Our goal is in general to control the shape of the aggregate demand profile. The operator of the DSM system (aggregator) can apply a steering signal to the root node to influence the demand profile. The steering signal gives an implicit description of the shape of the demand profile that we aim for (profile steering, which we describe

in Section 3.5, makes the shape explicit). We express the steering signal as an objective, which assigns a cost to an aggregate demand profile. The steering signal commonly takes the form of a price profile over time, but may e.g. also assign costs to the maximum demand value of the node. The goal of a node is to minimize cost according to the steering signal. Therefore, we consider the control of a node as an optimization problem. The demand profile of an internal (group) node depends on the profile of its children. The internal nodes generate new steering signals, and impose these on their children to control their demand profile. We often have to “demultiplex” the steering signal, i.e. translate it into smaller pieces, such that the children obtain signals that together lead to the desired demand profile. Children may get separate steering signals, although it is often more efficient to distribute the same signal to all children. Leaf nodes can be optimized directly. In the following, we refer to these cost based steering signals as a *pricing scheme*, which often (but not always) corresponds to a price profile.

3.2.2.3. PRICING SCHEMES AND PATTERNS

Although a pricing scheme influences the demand profile of a node, we do in principle not exactly know how the pricing schemes affects the demand profile. This would essentially make the pricing problem of the internal nodes (i.e. how to determine new steering signals for the children) a “blind” problem, even though the optimization within the children often already estimates the demand profile in response to the pricing scheme. We remove this source of uncertainty (which results from a lack of communication) by letting nodes specify the expected demand profile in response to the pricing scheme (however, the realized demand profile of devices may still be uncertain, and may be revised later on). We describe the expected demand profile more generally as a *pattern*. A pattern corresponds to the decisions that the child should take (or how the child should make decisions) to realize the demand profile, and is specific to that child node (including its subtree). In the hierarchical model tree, pricing schemes go downwards, and patterns go upwards. Note that not knowing the demand profile is not always a problem: if we only have to optimize the demand against a price profile, then we can optimize the group behaviour simply by passing these prices without actually knowing the demand profile.

3.2.2.4. MULTIPLE PRICING SCHEME OPTIONS

After an internal node has determined a pricing scheme for all children and received the corresponding patterns, we may add up these patterns (plus the demand of the node itself) to determine the demand profile of a node (including its children). To improve this demand profile, we allow a node to try out multiple pricing scheme options for its children. These pricing scheme options are usually determined iteratively, based on feedback from previous attempts. For example, we may see in the aggregate demand profile that the given pricing scheme for the children results in a large demand peak at a certain time in the future. We may (try to) reduce this peak by increasing the prices at that time. However, this may shift the peak to a different time, as each child may try to shift the demand in the same direction. Therefore, we allow the use of separate pricing schemes for children. Also, we may

combine the patterns of the children from different pricing scheme options. However, we can in general not (directly) combine different patterns, as this does not always lead to coherent decisions: for example, we can not “split up” two patterns of a single washing machine such that it starts for 50% at two different times (a lot of devices do offer this type of flexibility; see Section 3.4.5.1.4).

At some point, the root node of the tree is satisfied with the pattern that it produces, and commits to (the decisions corresponding to) this pattern for the time being.

3.2.2.5. INTERNAL COST

The shape of the aggregate demand profile is not the only thing that matters. In addition to the pricing scheme that is imposed by the parent of a node, nodes also consider their internal costs. For example, starting up a microCHP or imposing a heavy load on a transformer may lead to wear of these resources, and may thereby mean that we have to replace these devices earlier than if we would optimize without considering the shape of the demand profile. The extra costs are modelled as internal costs. We also count the consumables that are not accounted for in the demand profile (diesel, gas, water) and immaterial cost (discomfort, inconvenience, lost time) as internal cost. Note that possible (electric) efficiency losses are already accounted for in the demand profile. Furthermore, we may use the costs to specify soft constraints, which may be violated in exceptional cases (e.g. avoid cold showers unless this prevents damage to the neighbourhood transformer and no better options are available from other devices). Note that only *differences* in cost are relevant when we want to determine the least expensive option.

Similar to demand, internal cost flows upward in the hierarchical model. In contrast to the demand (which is represented as a vector over time), we represent the internal cost over the considered period with a single scalar value. To make the parent aware of the internal costs, we include this cost in the pattern, together with the costs of the patterns that we use from the children in the case of an internal node (which gives the internal cost of the subtree). The returned cost does not include the costs associated with the imposed pricing scheme to avoid double counting. For now, we only aim to minimize global cost, i.e. the cost from the imposed pricing scheme of the root node (from demand served outside the system boundary) plus the subtree internal cost of the root node, and assume that some mechanism can settle the costs between node owners.

3.2.2.6. EXAMPLE

To illustrate the structure of the model, we repeat an example that considers a balancing problem from Section 2.4 in Figure 3.2. At the top level, we have a VPP fleet control problem (at a system-wide “grid” level) that aims to follow a certain goal profile using a group of controlled devices (e.g. EVs). The devices are organized into households, which are in turn organized into neighbourhoods, which are called subgrids in the example to match the notation from [19; Chapter 4]. Note that we later try to avoid names such as “grid” and “subgrid” to avoid ambiguity. The root node (the grid) receives the goal profile as a steering signal (i.e. the pric-

FIGURE 3.2: Partitioned planning approach in TRIANA (repeats Figure 2.3).

ing scheme penalizes any distance from this profile), and translates this to pricing schemes for the subgrids. This may simply mean that we partition the profile proportionally over the children (e.g. if we have 10 children, each child has to realize $1/10^{\text{th}}$ of the demand profile in every time interval), even though this may be far from optimal. In turn, the subgrids determine pricing schemes for the households, and the households for their devices. In return, the devices give a pattern to their household, and the household adds up the device profiles from the patterns to determine the household demand profile. A household may then revise the device pricing schemes to improve the household demand profile. After this, the household gives its aggregate pattern to the subgrid controller, who may subsequently revise the household pricing schemes to reduce the distance to the target profile, until the subgrid controller is satisfied, after which it returns the subgrid pattern to the grid controller. The grid controller can do the same (i.e. repartition the goal profile over the subgrids [19: p. 70]), until we eventually produce the result pattern for the entire system.

To make a realization of the planning, we map specific optimization methods to the nodes in the tree of Figure 3.3, following the approach from [19: Chapter 4]. At the top (grid controller) level, we receive a goal profile, so the root node has to accept a goal profile as (input for) a pricing scheme. The specific interpretation of the goal profile in terms of objective cost is node implementation specific. We use a partitioning node as the root (“grid”) node, which distributes the goal profile over the subgrid controllers below it. For the subgrid controllers, we choose to use IDDP (Section 3.3), which translates a goal profile into price profiles for the households. IDDP interprets the goal profile as upper and lower bound constraints on the demand profile, and gradually adapts the prices for the households to push the profile within these bounds (higher prices at times where the demand is too high, and lower prices at times where the demand is too low). These prices lead to an independent cost function at the household level, for which we can determine the optimal solution simply by passing on these prices to the devices, and then adding up the resulting device profiles (see Section 3.2.5.3). At the bottom level, we solve the device problems by *dynamic programming* (DP).

FIGURE 3.3: Realization of planning hierarchy in TRIANA corresponding to Figure 3.2 (repeats Figure 2.4).

We have quite some freedom in the configuration of the realization of the problem: for example, Bakker [19: Figure 4.4] puts an IDDP node in the root position, and places pass-through nodes at the subgrid level. As [19: Figure 4.9] illustrates, this approach leads to much less smooth demand profiles than the example approach. Considering that the reference uses 4 subfleets of 50 households each, we would have expected even less smooth results in the latter approach. We also could have used partitioning nodes all the way down to the household level, which is comparable to the approach that we follow in Section 3.5. The DPs may be replaced with other optimization methods, e.g. a MIP or a problem specific solution method, or use a fast-but-inaccurate heuristic in the first iterations of the optimization process. Note that the effective use of this configuration flexibility requires a flexible software implementation (earlier work considered only a few hard-coded configurations).

In the following, we present the node model more formally.

3.2.3. NODE MODEL

In this subsection, we make the model that we outlined in Section 3.2.2 more formal, such that we can give a unified presentation of the optimization methods in the rest of this chapter, and allow the interchangeable use of the methods (where this is possible).

3.2.3.1. BASIC NODE STRUCTURE

The basic element of the model is a node. A node can be e.g. a device, a heating system, a house, or a complete neighbourhood. The general elements present in a node are shown in Figure 3.4, together with two child nodes; we introduce these elements in the following. The environment can influence a node by imposing a pricing scheme (i.e. a steering signal). In return, the node estimates the (optimized) response to this pricing scheme. We index a node by $\mathcal{N}$, and let $\mathcal{N}$ be the set of all nodes in the system at hand.

FIGURE 3.4: General node x with children.

As in earlier models, we start with a time discretization. We split up the optimization horizon in n_t time intervals, each of a fixed length τ , and denote the set of time intervals as $\mathcal{T} = \{1, \dots, n_t\}$. A node x has a demand vector $\mathbf{x}_x^{\text{int}} = \langle x_x^{\text{int}}(1), \dots, x_x^{\text{int}}(n_t) \rangle$, which contains variables that describe the average demand (i.e. power) for all time intervals (we discuss the superscript int shortly). Negative demand represents supply, and cancels out demand. Furthermore, a node has an internal cost z_x^{int} over the considered horizon (a single scalar value).⁴

3.2.3.2. HIERARCHICAL STRUCTURE

Nodes are placed in the context of a hierarchical tree structure. Every node is at the top of a subtree, which in the case of a leaf node is just itself (e.g. a device). The internal nodes of the tree have children. We denote the set of children of a node x as $\mathcal{I}_x$. We may use a “path notation” to refer to a specific child where this is unclear from the context, e.g. $\mathbf{x}_{x:1:2}^{\text{int}}$ gives the demand of the second child of the first child of x .

3.2.3.2.1. Structure of demand In this tree, nodes contribute demand to their parents. This recursively gives the aggregate demand at the root node of the tree. The aggregate demand of a node x including its descendants, which we call the subtree demand, is denoted as $\mathbf{x}_x$. The node itself contributes the internal demand $\mathbf{x}_x^{\text{int}}$ to the subtree demand. In the case of an inner node, the internal demand gives all demand that is not accounted for at a lower level (e.g. local distribution losses). The demand of the descendants, which we call the downward demand, is denoted

4. We may in principle also represent costs as a sequence of discrete cost events and continuous cost intervals, which may be useful in a rolling horizon approach, yet leads to a more complex model.

as $\mathbf{x}_\mathcal{N}^{\text{down}}$. The downward demand is equal to the subtree demand of the direct children of $\mathcal{N}$. Together, this gives:

$$\mathbf{x}_\mathcal{N} = \mathbf{x}_\mathcal{N}^{\text{int}} + \mathbf{x}_\mathcal{N}^{\text{down}} \quad (3.1)$$

$$\mathbf{x}_\mathcal{N}^{\text{down}} = \sum_{i \in \mathcal{I}_\mathcal{N}} \mathbf{x}_i. \quad (3.2)$$

3.2.3.2.2. Structure of cost A node is responsible for the optimization of its own subtree, including the children, and can influence the behaviour of its children by determining pricing schemes for them. From an implementation perspective, the parent node “adopts” its children, but conceptually it is more clear to keep the children as separate nodes. The operator of the system, e.g. an aggregator, imposes a pricing scheme on the root node of the tree.

To optimize the subtree of $\mathcal{N}$, we have to consider the local view of cost $z_\mathcal{N}$. This cost consists of an upward cost $z_\mathcal{N}^{\text{up}}(\mathbf{c}_\mathcal{N}, \mathbf{x}_\mathcal{N})$ and a subtree cost $z_\mathcal{N}^{\text{sub}}(\mathbf{x}_\mathcal{N})$.

The upward cost $z_\mathcal{N}^{\text{up}}$ gives the cost to satisfy the demand (residue) $\mathbf{x}_\mathcal{N}$, where $\mathbf{c}_\mathcal{N}$ gives the pricing scheme that is imposed on $\mathcal{N}$ by its parent (note that if we take the perspective of the parent $\mathcal{N}'$ of $\mathcal{N}$, $\mathbf{c}_\mathcal{N}$ is equivalent to $\mathbf{c}_i$ for $\mathcal{N} = i \in \mathcal{I}_{\mathcal{N}'}$). We omit the arguments $\mathbf{c}_\mathcal{N}$ and $\mathbf{x}_\mathcal{N}$ where the considered pricing scheme and demand vector are clear. In this context, upward means that the supply source is either the node directly up in the hierarchy, or outside of the system boundary. The pricing scheme may be a simple linear price vector over time ($z_\mathcal{N}^{\text{up}} = \tau \cdot \mathbf{c}_\mathcal{N}^\top \mathbf{x}_\mathcal{N}$), but may also represent more complex pricing schemes. We also use the notation $\mathbf{c}_\mathcal{N}^\ell$ to refer either to a pricing scheme that consists only of a linear price vector, or to the linear price vector component within more complex pricing schemes. Note that there is no analog to upward cost for the demand vector.

The subtree cost $z_\mathcal{N}^{\text{sub}}$ consists of the internal cost $z_\mathcal{N}^{\text{int}}$ and the downward cost $z_\mathcal{N}^{\text{down}}$, and is analog to $\mathbf{x}_\mathcal{N}$. The internal cost $z_\mathcal{N}^{\text{int}}$ may represent e.g. the wearing costs of a device or a valuation of inconvenience. The internal cost of internal nodes may also consider the demand vectors of specific groups of children, e.g. the maximum demand sum of the children in a specific branch of the grid. The downward cost $z_\mathcal{N}^{\text{down}}$ describes the cost that the chosen pricing scheme imposes on the descendants of $\mathcal{N}$. Note that $z_\mathcal{N}^{\text{sub}}$ does not include the upward cost. More formally, this gives:

$$z_\mathcal{N} = z_\mathcal{N}^{\text{up}} + z_\mathcal{N}^{\text{sub}} \quad (3.3)$$

$$z_\mathcal{N}^{\text{sub}} = z_\mathcal{N}^{\text{int}} + z_\mathcal{N}^{\text{down}} \quad (3.4)$$

$$z_\mathcal{N}^{\text{down}} = \sum_{i \in \mathcal{I}_\mathcal{N}} z_i^{\text{sub}}. \quad (3.5)$$

At higher levels (in the ancestors of $\mathcal{N}$), we should account for the subtree costs that result from imposing some pricing scheme (a different pricing scheme may lead to different internal costs). For the accounting of the costs of this node at a higher level, we should use $z_\mathcal{N}^{\text{sub}}$ to avoid double counting, except in the root node where we should consider $z_\mathcal{N}$ as otherwise the external costs (for the aggregate demand profile at the root node) are not accounted for.

3.2.3.3. CONTROL STRATEGY

The demand vector depends on the decisions that the node makes in response to the pricing scheme. We let the behaviour of a node depend on the minimization of the local view of cost $z_{\mathcal{X}}$ that we discussed in Section 3.2.3.2.2. For uniformity, we only consider minimization problems; maximization problems are covered by negating the objective. The minimization may be approximate.

In principle, the outcome of this minimization is a control strategy $\pi_{\mathcal{X}}$ that describes how node $\mathcal{X}$ makes decisions over time. The demand vector $\mathbf{x}_{\mathcal{X}}(\pi_{\mathcal{X}})$ results from using this strategy. The strategy should not have any direct dependencies on other nodes, but may be parameterized by values that are determined dynamically for which we can estimate the dynamics. For internal nodes in the hierarchy, choosing a strategy $\pi_{\mathcal{X}}$ implies the choices of the strategies π_i of its children $i \in \mathcal{I}_{\mathcal{X}}$, and recursively of all children in the subtree below it. Note that this description contrasts with the earlier work on TRIANA, which considers $\mathbf{x}_{\mathcal{X}}$ directly as the outcome of the planning procedure. We may provide a more accurate representation by simulating the use of the strategy to determine the expected demand and cost (Chapter 5 uses this approach). The use of a control strategy may result in multiple demand vector realizations, as is further explored in Chapter 4; in this chapter, we for now assume that the values correspond to the expected average value of the demand in each time interval. Note that we generally choose to leave this parameter $\pi_{\mathcal{X}}$ implicit to limit notation overhead.

3.2.3.4. PRICING SCHEMES AND PATTERNS

A parent node $\mathcal{X}$ is responsible for the behaviour of its children, but can not control or optimize the children directly. To support spatially decoupled planning, the parent node can impose a pricing scheme $\mathbf{c}_i$ on each of its child nodes $i \in \mathcal{I}_{\mathcal{X}}$, and poll how the child expects to behave under a given assignment of $\mathbf{c}_i$.

The response p to an assignment of $\mathbf{c}_i$ is specified as a *pattern*, and gives the result of the strategy that the child uses in response to $\mathbf{c}_i$. A pattern represents a feasible assignment of all parameters in the subtree of i . More precisely, a pattern p consists of the expected demand values $\mathbf{x}_{i,p}$, and the cost $z_{i,p}^{\text{sub}}$ of the subtree for which the child node i is responsible, including itself. We usually denote a specific pattern e.g. as $\langle \mathbf{x}_{i,p}, z_{i,p}^{\text{sub}} \rangle$. The definition of a pattern has been chosen such that it is independent of the pricing scheme that we have imposed on the child, i.e. the parent may locally evaluate the patterns with a different pricing scheme. Although the patterns currently only describe the demand and the related cost, we may later extend it to represent other externally relevant properties of the node, e.g. spare flexibility and demand risk. A pattern is specific to a single node (and is coupled to the patterns of its descendants, who have provided patterns as well). We sometimes use a node subscript when it is not clear to which node a pattern belongs, e.g. p_i .

A parent may try out multiple values for $\mathbf{c}_i$. Furthermore, a single request with a pricing scheme $\mathbf{c}_i$ may in principle even result in multiple patterns, which all represent possible responses of i (which do not have to be optimal with respect to $\mathbf{c}_i$, except when the upstream optimization algorithms rely on optimality). We represent the patterns that have been generated by a child i with a pattern set $\mathcal{P}_i$. Note

that $\mathcal{P}_i$ may not contain all generated patterns, as the set may be pruned. Most methods even maintain only the last (or the best) generated pattern solution in $\mathcal{P}_i$.

Although it is easy to partition the model along the boundaries of the nodes in the model, and introduce the concept of pricing schemes to relate the parts, it is not immediately clear what the prices should look like to find a meaningful planning for the combination of parts, unless the problems are independent (e.g. a cost minimization against linear prices $\mathbf{c}_x^l$ with only device-level constraints). Therefore, we need a structured approach by which we can not only split up problems, but also put them back together effectively. Pricing schemes and pattern management methods make up the heart of this chapter.

3.2.3.5. NUMERICAL EXAMPLE

To illustrate the flow of demand and cost in the hierarchical model, we give a numerical example in Figure 3.5. The example considers a neighbourhood with two households, each with two devices. We choose a time interval length $\tau = 1$ h and express prices in €/kWh to simplify the calculations in the example. As discussed earlier, prices first flow all the way down, and then patterns flow upwards. We present only a single pattern for each node. Later in this chapter, to simplify the notation and the figures, we do not make the pricing scheme and response pattern arrows explicit, and instead draw just lines, each of which corresponds with a downward imposed pricing scheme arrow and an upward response pattern arrow.

At the top level, we have a simple linear pricing scheme $\mathbf{c}_x$. This pricing scheme is transformed by the root node x into pricings $\mathbf{c}_{x,1}$ and $\mathbf{c}_{x,2}$ for the households. The transformation (price increase of a few cents) aims to account for the grid losses that are caused by the households in x , and may also account for wear on the grid (not modelled here; this could give internal cost in x). The households pass on these prices unchanged to the devices (a household is a pass-through node; see Section 3.2.5.3), who then optimize their local view of cost according to these prices. This results in a demand profile and an internal cost, e.g. $\mathbf{x}_{x,1,1}^{\text{int}}$ and $z_{x,1,1}^{\text{int}}$. A device has zero downward cost and demand as it has no descendants, and therefore also $\mathbf{x}_{x,i,i'} = \mathbf{x}_{x,i,i'}^{\text{int}}$ and $z_{x,i,i'}^{\text{sub}} = z_{x,i,i'}^{\text{int}}$ for $i \in \mathcal{I}_x$ and $i' \in \mathcal{I}_i$. Each device result is represented by a pattern $\langle \mathbf{x}_{x,i,i'}, z_{x,i,i'}^{\text{sub}} \rangle$, which is given to the corresponding household.

In the example, the upward costs to buy electricity are dominant. The devices all try to shift demand away from the third and fourth time interval where electricity is expensive. Note that if all devices shift their demand to the cheapest time interval, then this may lead to a demand peak at that time, which may be countered by a price increase there. Device 1 of household 1 appears to be unable to shift the demand from the third time interval (it may have to run for three consecutive hours, e.g. such as a washing machine), and device 1 of household 2 appears to be unresponsive altogether. It could also be that the internal costs to shift the demand away from these time intervals is too high (when the upward costs are very high, we may be willing not to run the washing machine at all if the reward/the user's internal cost is high enough). Device 2 of household 2 incurs a relatively high internal cost in the optimal solution (e.g. to start using the device earlier than we prefer to).

FIGURE 3.5: Numerical example of hierarchical model ($\tau = 1 \text{ h}$, $n_t = 5$).

In the households, we add up the patterns from their devices to produce the household patterns. A household has zero internal demand and internal costs (in the case at hand); the household demand is therefore determined only by the downward demand and costs (the sum of the subtree demand and costs of its children). Note that due to the linear structure of the problem, the upward cost of the household node, which is computed based on the household demand (e.g. $\mathbf{x}_{\mathcal{N}:1}$), equals the sum of the upward costs of the devices below them. Also, the local view of cost of a household equals the sum of the local view of cost of its children (e.g. $z_{\mathcal{N}:1} = z_{\mathcal{N}:1:1} + z_{\mathcal{N}:1:2}$).

The neighbourhood node adds up the patterns from the households, leading to $\mathbf{x}_{\mathcal{N}}^{\text{down}}$ and $z_{\mathcal{N}}^{\text{down}}$. Although this is not shown in the example, the neighbourhood node may combine patterns from households that are generated with different pricing schemes. As noted earlier, the demand of the households leads to losses, which are represented as internal demand in $\mathcal{N}$. The node does not have internal costs. Together with the downward demand and costs (the subtree costs of the households), this gives the aggregate/subtree demand $\mathbf{x}_{\mathcal{N}}$ and cost $z_{\mathcal{N}}^{\text{sub}}$ of $\mathcal{N}$. The total cost is determined by the local view of cost of the root node $z_{\mathcal{N}}$, which consists of the cost that we incur outside of the system (above the root node), i.e. the upward cost $z_{\mathcal{N}}^{\text{up}}$ of the root node, and the total internal costs $z_{\mathcal{N}}^{\text{sub}}$ that we incur in the tree (the “subtree” of the root node $\mathcal{N}$).

Earlier, we derived pricing schemes $\mathbf{c}_{\mathcal{N}:1}$ and $\mathbf{c}_{\mathcal{N}:2}$ from the root node pricing scheme $\mathbf{c}_{\mathcal{N}}$ (all taking the form of price vectors in the example), which aim to compensate for the losses in $\mathcal{N}$. We see that the $z_{\mathcal{N}:1} + z_{\mathcal{N}:2} = €4.44$ is larger than the actual cost $z_{\mathcal{N}}$, indicating that we put too much weight on the problem of $\mathcal{N}$, which may lead to solutions with a too high internal cost in the descendants. Therefore, we should lower the prices. Note that this does not yet tell *which* prices should be lowered (we should look deeper at the structure of the problem to determine this). We may try out several pricing schemes for the households (not illustrated).

At the end of the optimization, we commit to a specific pattern $p_{\mathcal{N}}$, which has a demand $\mathbf{x}_{\mathcal{N},p_{\mathcal{N}}}$ and a subtree cost $z_{\mathcal{N},p_{\mathcal{N}}}^{\text{sub}}$. Outside of the system boundary, the total cost becomes $z_{\mathcal{N},p_{\mathcal{N}}} = €4.30$. Most of this cost comes from the external cost to buy electricity, and 10.2% comes from internal costs in our tree (mostly from device 2 in household 2).

3.2.4. MULTICOMMODITY/HYBRID ENERGY MANAGEMENT

3.2.4.1. INTRODUCTION

DSM resources usually have a limited flexibility (we may not postpone demand indefinitely), and often have a more or less fixed energy content. This may lead to high forced demand peaks at some critical times, and makes it difficult to provide dynamic demand response without compromising the planned behaviour in the future (a lower demand now leads to a higher demand in the future). A multi-commodity approach may provide a possible answer to these problems. Some commodities are less time sensitive than electricity, and may be stored over longer periods. Furthermore, transport peaks in the gas distribution grid may be less of a problem than with electricity. This applies in particular to gas, or when the

commodity is stockpiled locally (e.g. oil or gas containers). PHEV also have this hybrid energy capability: we may choose not to fully charge a car and run on fuel instead. Section 5.5.3 considers an use case where we have a hybrid heat pump configuration, which can produce heat either by using a gas burner, or by using an electric heat pump. The use of these commodities usually comes at a higher cost, as otherwise we may would always choose to use the non-electric option, and requires the maintenance of parallel infrastructures. However, volatile electricity prices and the avoidance of electricity infrastructure investment costs may still make this an attractive alternative.

In related work on DSM, multicommodity optimization is considered as a challenging problem, because the consumption from multiple sources has to be optimized together. This leads to a combinatorial auction, which excludes some common solutions such as the prevalent single commodity Walrasian auction method (Section 2.3.2.1.5). Two workarounds are to consider the costs for the commodities to be static (even though it may vary over time), or to shift the focus to a different commodity (e.g. HeatMatcher [268]), both of which are not always appropriate. We observe that our optimization model from Section 3.2.3 already gives a combinatorial auction problem, as we have to make coupled decisions for the time intervals. In this subsection, we present an orthogonal extension to the model of this section to support multicommodity energy stream management. Earlier work on TRIANA supported multiple commodities only in the control phase (Section 2.4.2.1), and not in the planning phase. We use this extension in Chapter 5.

3.2.4.2. MODEL

We extend the demand model of Section 3.2.3.2.1 with a commodity dimension $c \in \mathcal{C}_{\mathcal{N}}$, which is expressed with a subscript c , e.g. $\mathbf{x}_{\mathcal{N},\text{el}}$ gives the electricity demand vector of a node $\mathcal{N}$ ($c = \text{el}$). For unspecified commodities, the demand is implicitly $\mathbf{0}$. The patterns consider a coupled commodity consumption (e.g. the gas consumption, electricity production and heat production of a CHP are coupled). Typically all nodes use the same commodity set $\mathcal{C}$. The demand values should represent a flow rate, e.g. an energy flow rate/power value (in W) or volumetric flow rate (in m^3/s). Depending on the context, when we leave out the commodity dimension, operations should be applied to all commodities. When we do not make the commodity set explicit, we use $\mathcal{C} = \{\text{el}\}$, i.e. only electricity. The commodities should all be delivered by the same parent (we discuss this further in Section 3.2.4.4).

The pricing scheme from Section 3.2.3.2.2 does in principle not have a commodity dimension as it may also consider combinations of commodity consumption (the different commodities may be interdependent). However, the cost functions are commonly (presented to be) independent between commodities, and in this case we do present the pricing scheme with a commodity subscript, e.g. $\mathbf{c}_{\mathcal{N},\text{gas}}$ for the gas pricing scheme. For unspecified commodities, the pricing scheme gives an infinite cost (∞) for a nonzero demand, and 0 otherwise. It should be clear that the upward cost is then determined by the sum of the costs for each of the commodities. In cases where the pricing scheme contains a norm, e.g. the maximum value, we generally should compute these norms separately for each commodity.

The cost model remains the same and is in principle not split up by commodity. However, in the case of where the commodity pricing schemes are independent, we may split out the upward cost by commodity source as in Section 5.5.3.

Although we describe $\mathcal{C}$ as a set of commodities, we could also view it as a set of ports, where the concept of commodities simplifies the hierarchical connections between ports (the model of Section 2.4.2.1 supports more general connections in the form of *streams*). A node may translate between commodities, e.g. we may select an electricity port from a three-phase electricity connection with ports L1/L2/L3 (this type of flexible modelling has been left for future work).

In the rest of the chapter, we choose to omit the commodity dimension from the definition of the optimization methods, as adding this dimension is mostly an orthogonal operation. Leaving out this dimension simplifies the notation.

3.2.4.3. FURTHER CONSIDERATIONS

There are some practical considerations that should be addressed to apply the multicommodity model from Section 3.2.4.2. We address these in the following.

3.2.4.3.1. *Scaling of coefficients* An important consideration in multicommodity optimization problems is the scaling of coefficients between the different commodities. For example, we may measure electricity consumption in W and gas consumption in m^3/s , which typically gives coefficients in a completely different order of magnitude. Where we can typically simply round values in W to integer values (few people care about power values smaller than 1 W), this is inadequate for consumption values with extremely large or small values, and we may have to use an appropriate prescaling to put the numbers in a workable range. If the constant refers to a quadratic term, then the scaling should be squared as well, i.e. for a constant α , a scaling β and a variable x , we have $\alpha(\beta x)^2 = \alpha\beta^2 x^2$. Even if coefficients are expressed in the same unit, there may still be large differences between the coefficients, e.g. gas (one of the goals of multicommodity control is to exploit these differences).

3.2.4.3.2. *Different physical behaviour* We model the multicommodity support in the same way as the electricity support, i.e. as energy streams in a bottom-up demand model. This may for some cases lead to inaccurate results. For example, the physical properties of the energy exchange are far more important in heat grids, which should also account for temperatures, flow restrictions, and thermal inertia in the infrastructure. In other cases, the scheduling method may be “over-engineered”: e.g. for gas, we may care about the total demand volume over the day (or over the year), and are less interested in the balancing of the demand over time within a day. Depending on the predictability of the demand, a simple time (of day) invariant linear pricing scheme may have been sufficient. We may also consider to use a longer time interval length for the commodities that are less time sensitive (however, this makes the implementation more complex).

3.2.4.3.3. *Efficient evaluation in optimization problems* Multicommodity problems add an extra dimension to the optimization problem, and may thereby make the optimization more challenging. Usually, this dimension is small in practice (1 – 3 commodities). In group optimization problems, the extra dimension usually adds a linear cost (to consider the extra vectors), and may slow down the convergence of the search as we have to settle more pricing schemes. In the device optimization problems, in most cases it is possible to incorporate this dimension with only a simple preprocessing operation, such that it does not contribute to the complexity of the search. These operations do assume that the pricing schemes are independent between time intervals, and are focused on a DP approach (but may be useful elsewhere as well).

If we have a limited (and discrete) number of commodity consumption options, then we may set up a *lookup table* (LUT) of cost contribution values. That is, we evaluate the costs once for every option in every time interval, and then use these precomputed values throughout the search. In terms of search complexity, the multicommodity aspect typically does increase the number of available control options (and thereby the branching) within the search.

If we have continuous coupled consumption options with a fixed consumption ratio between commodities (or multiple of these options), then we may introduce a proportional auxiliary variable, and then express the cost in terms of this auxiliary variable. This is similar to the approach that is used within the energy stream model of Section 2.4.2.1, e.g. their microCHP has a (steady-state) fixed multiplicative ratio $10\text{ W} : -1\text{ W} : -8\text{ W}$ on the range $0 - 1000$ in gas consumption, electricity production and heat production (see also [231: p. 45]), although their approach only considers a linear cost model. For linear and quadratic (or in general polynomial) pricing schemes, this transformation is trivial. Note that the same approach can still be used if some of the commodities have an offset from 0, or some other well-defined transformation for which we can determine the combined cost function in advance (the ratio becomes nonlinear in this case). The method should also be applicable for most cost functions that are interdependent between commodities, as long as we can determine the combined cost function in a single controlled variable (within a time interval).

3.2.4.4. INDEPENDENT COMMODITY PROVIDERS

In a multicommodity context, scenarios may occur with separate commodity providers, leading to a situation where nodes in our network may have multiple parents, one for each relevant commodity. However, from a control perspective this is not desirable. If in such a situation the pricing scheme for a commodity is given on beforehand or out of our control (we are price takers, or may estimate our influence on the price locally), then we only need a single controller, and the considered problem does not occur. Otherwise, a situation may occur with a tug-of-war between the parents, where each chooses prices according to their own interests. For example, if we consider a hybrid heat pump that can consume either gas or electricity, then the gas provider may minimize costs if the device completely runs on electricity, and the electricity provider may minimize costs if the device runs completely on gas (or the converse if we account for sale profits). If the parent

nodes that are responsible for the commodities can get along and work in the best interest of the child node at hand, then we may address the problem by combining the parent problems. Otherwise, we have a kind of competitive problem, where we have to guess how the other parent(s) price their commodities, and may at best settle for a Pareto optimal solution. The problem here is that the parents have to agree on the pattern response of their common children, and to this end the parents should agree on the pricing schemes of the commodities.⁵ In each of these cases, we should “splice” the pricing schemes and response patterns according to the commodities that are relevant (i.e. pack and unpack the pricing schemes and demand values for each of the parents, according to what parent provides which commodity).

We may address this problem as follows. We put the parents together in a new node. If the parents themselves again have separate parents, then these should be put together as well, and so on. All “upward” edges now lead to combined problems, such that we do not choose the patterns for the commodities separately. The parents still may determine their pricing schemes for the commodities separately, but must eventually agree on the chosen pattern, as well as on the prices (as different prices usually lead to a different pattern). Although we can now relax the strict “single parent” relation, we still require a DAG hierarchical relationship, i.e. the nodes must be a partially ordered.

3.2.4.5. EXPLOITING MULTIPLE COMMODITY PROVIDERS

A comparable problem as in Section 3.2.4.4 occurs if we have multiple providers for the same commodity, and we can choose how much we consume from each provider. In this case, we may resolve the conflict in the same way, i.e. the providers should agree on the demand pattern and on the cost functions. To optimize the use of commodities from these providers, locally we should now minimize (in the case of two providers) a composed cost function $z(x_\alpha) + z(x_\beta)$ with $x = x_\alpha + x_\beta$, where x gives the demand for the commodity at hand, and x_α and x_β give the demand that we assign to provider α and β , respectively. The composed cost function may need some extra constraints to prevent “cheating”, e.g. to prevent that we sell energy from one provider to the other ($x_\alpha, x_\beta \geq 0$). Note that the same reasoning may apply in some cases where we can choose between buying two different commodities. If there are more than two possible providers, then we may compose the cost functions recursively.

For a simple linear cost function, the “composition” implies that we consume everything from the cheapest provider (without extra constraints or asymmetric tariffs, we would furthermore try to sell as much as possible from the cheap provider to the more expensive provider).

For a quadratic cost function (without restrictions), composition is possible as well, but the procedure is a bit more involved than in the linear case. The result is again a quadratic function. We present this method in Appendix C.1.1 (p. 307).

5. In principle, it is also sufficient if the parents only agree on the response pattern, as the response of the child may be (partially) indifferent to the exact price values, but this may give challenges if we later on decide to make changes to the prices. Conversely, the same prices may give different responses.

3.2.5. SCALABILITY

3.2.5.1. INTRODUCTION

In the first part of Section 3.2, we have proposed a scalable hierarchical model for energy systems that allows the specification of very large (e.g. worldwide) energy systems. However, even if we can specify such large systems, this does not always lead to a workable system. It is commonly assumed that the partitioning of a problem automatically leads to a scalable solution method. We demonstrate in this subsection that this belief does not hold if we consider nested optimization problems, unless we take some precautions. The reason for this is that the iterative request/response interaction method, as introduced in Section 3.2.2.4, leads to an exponential growth in the amount of work towards the bottom of the hierarchy. In this subsection, we illustrate the scaling behaviour of nested optimization routines, independent of the specific method used. Furthermore, we propose model transformations to modify the structure of the problem tree, which may help to improve scalability. We experiment with different tree structures later in Section 3.4.4. Instead of the typical *divide and conquer* idea that motivates to split up a problem as often as we can, the results of Section 3.2.5.2 and Section 3.4.4 favour configurations where the internal nodes have as many children as we can manage.

3.2.5.2. NESTED COMPLEXITY

The hierarchical model of this section suggests a nested optimization approach, which introduces a scalability hazard. Typically, every internal node uses an iterative process to optimize its cost. If we have a tree with multiple levels, then these iterative processes are nested as well. As the height of the tree increases, this leads to an exponential growth in work at the bottom nodes of the tree. This contrasts with typical scalability issues, which tend to occur towards the top of the tree (these may occur as well if we use an event triggered replanning approach as discussed in [19: Section 4.4], which may lead to linear increase in the number of events with the number of children unless we can keep the event response local). The bottom nodes are generally implemented on much less powerful hardware than the nodes at the higher levels for cost and efficiency reasons, even though in a naive approach these have to carry the brunt of the work. The complexity of the search is exponential in the height of the tree, where the iteration count n_k determines the base and the tree height n_d determines the exponent, i.e. nodes at the bottom of the tree have to be optimized $(n_k)^{n_d}$ times in a direct implementation. However, the number of devices that we can support is exponential in the height of the tree as well: if every internal node has n_g children, then we may support $(n_g)^{n_d}$ devices. If $n_g \gg n_k$, then the approach may scale sufficiently (but not infinitely). Note that we may ignore the actual number of devices at the bottom of the tree, as the distributed computation power is in general proportional to the number of devices. We see no need for the foreseeable time to scale beyond a billion devices, yet it is realistic to consider a million controllable devices within the next 10 years (e.g. when EVs become commonplace).

FIGURE 3.6: Exponential growth of iteration count in tree height with 1000 000 device nodes at bottom level for $n_k = 3$ and $n_k = 10$.

To make this more concrete, we consider an example for the scaling behaviour if we want to support a large group of $n_{\text{dev}} = 1000\,000$ devices, e.g. a national scale VPP, which we illustrate in Figure 3.6. In this example, we split up the problem in groups of $n_g = 100$ children (Figure 3.6a) or 1000 children (Figure 3.6b), which gives a (minimum) tree height of 2 and 3, respectively. Note that this assumes that the levels are balanced, which is not always true. For example, it is common practice to introduce a node at household level that handles only a few devices (however, this node benefits from locality of reference as in-house communication is relatively cheap). More generally, $n_d \geq \lceil \log_{n_g} n_{\text{dev}} \rceil$. Although this difference between $n_d = 2$ and $n_d = 3$ seems small, exponential growth is relentless. For example, if we have to spend $n_k = 10$ iterations at each level in a tree of height 3, then we spend 1000 iterations at the bottom level, whereas in a tree of height 2 we would only spend 100 iterations. A lower iteration count leads to less extreme differences, e.g. 9 against 27 iterations in the example.

The above motivates the use of nodes with a large number of children. However, this approach has disadvantages, in particular towards the bottom of the tree. If an iteration in a group optimization problem is computationally expensive (e.g. an algorithm that is quadratic in the number of children or a MIP), then for the lower nodes it may start to dominate the exponential process below it, especially as the bottom level problems can be solved in parallel. Furthermore, the communication with a large number of children from a central point may be impractical. A large number of children also makes the node more vulnerable as a single point of failure, although failover mechanisms can help to cope with this.

There are some methods to keep the iteration count limited in order to keep the growth in work tolerable. First, there are substantial differences between the solution methods and their configuration in the number of iterations that we need to find an acceptable solution, ranging from a single to hundreds of iterations. We discuss a special case where we can obtain an optimal solution in a single iteration in Section 3.2.5.3. Choosing a suitable method that fits to the problem at hand is therefore the first step. Within the optimization methods, there are many opportunities to reduce the iteration count. For example, the repetitive nature of the planning process, often using only slightly changed parameters, gives opportunities for the reuse

Input: imposed pricing scheme $\mathbf{c}_x$ (must be linear)

Output: pattern (set) for x which assigns one pattern p_i for all $i \in \mathcal{I}_x$

- 1 · $\mathcal{P}_i \leftarrow$ solve i with prices $\mathbf{c}_i = \mathbf{c}_x$ for $i \in \mathcal{I}_x$ # in parallel
- 2 · return $\left\{ \left(\mathbf{x}_x = \mathbf{x}_x^{\text{int}} + \sum_{p_i \in \mathcal{P}_x} \mathbf{x}_{i,p_i}, z_x^{\text{sub}} = z_x^{\text{int}} + \sum_{p_i \in \mathcal{P}_x} z_{i,p_i}^{\text{sub}} \right) \mid \mathcal{P}_x : \prod_{i \in \mathcal{I}_x} \mathcal{P}_i \right\}$

ALGORITHM 3.1: Pass-through group optimization problem.

of earlier results (in particular the patterns). With this reuse and a low iteration count, the child problems start to look more as communicating rather than subordinate problems. A further technique is local preprocessing, where the bottom nodes already try to work towards the goals of the nodes above them. For example, the lower nodes in the tree may already try to make a peak shaved profile, which reduces the effort (iterations) that we have to spend at the top level. Extending on the earlier idea of pattern reuse, as a side effect this warms up their pattern sets, which may greatly reduce the need to request patterns from the subproblems below it. These ideas have been (partly) explored in the experiments of Section 3.4.4.5.

3.2.5.3. PASS-THROUGH NODE

If the subproblems of a node x do not interact, then we can significantly simplify the optimization procedure, and avoid the exponential growth of work towards the bottom of the problem tree that we discussed in Section 3.2.5.2. In particular, we consider the case where we do not have internal cost and demand, or where this demand $\mathbf{x}_x^{\text{int}}$ and cost z_x^{int} is independent from the children (and can therefore be optimized separately). Furthermore, the imposed pricing scheme should be linear (or zero), and x should not introduce any further constraints. In this case, we may simply redistribute the pricing scheme to all subproblems, i.e. $\mathbf{c}_i = \mathbf{c}_x$ for all children $i \in \mathcal{I}_x$. Due to the independence of the subproblems, we only have to solve each child subproblem once for a given value of $\mathbf{c}_x$, and the sum of these patterns provides an optimal assignment of the subtree demand $\mathbf{x}_x$ and cost z_x^{sub} (provided that the patterns from the children are optimal). To determine the outcome, we can add up the patterns from the children (and, if applicable, add the internal demand and cost). The independence of the problems avoids an iterative procedure, and thereby the recursive growth of the number of bottom-level iterations that we discussed in Section 3.2.5.2. This case is very common in TRIANA, and results in a significant decrease in computational effort towards the bottom of the problem tree. However, we later see in Section 3.4.4.3 that this approach may give extra iterations at a higher level. A comparable procedure may be developed if the internal demand and cost are a linear combination of the demand and cost values (e.g. 5% losses, although this also leads to a gain for negative values). In this case, we have to transform the prices, but still only have to evaluate these prices once.

Algorithm 3.1 gives a more formal description of this procedure. In line 1, we send $\mathbf{c}_x$ to all children $i \in \mathcal{I}_x$, and we gather the corresponding solutions $\mathcal{P}_i$ from each child. Next, we combine the subproblem solutions into a solution for x . This is

complicated by the possibility that children can produce more than a single pattern. We return all possible combinations of patterns (i.e. a Cartesian product) in line 2. Within line 2, $\mathcal{P}_x$ assigns one pattern to each i from $\mathcal{P}_i$. Note that we may also first select the lowest-cost pattern from each set, which leads to a single solution.

Next to the more algorithmic improvements proposed above, we may also control the structure of the problem tree. In the following, we discuss transformation by which we can (in many cases) change the shape of the problem tree, such that it becomes more tractable for the optimization methods.

3.2.5.4. TREE TRANSFORMATIONS

The most natural representation of a problem in our model does not always lead to the most efficient implementation for the problem. Following the discussion in Section 3.2.5.2, the natural representation may give nodes that have either too few children (which leads to a deep problem tree), or too many children (which leads to operational challenges and hard problems for the internal nodes). Therefore, we may transform the problem tree to cater for the needs of the problem or the solution methods at hand. In some cases, we can adapt the shape of the tree without changing the objective and constraints of the problem. In other cases, the transformation of the tree may change the meaning of parts of the problem. For example, different interpretations of following a profile may be used depending on the concrete setting. This impact may in some cases be intentional: for example, Hoogsteen [160] groups the demand of households on the same feeder and phase in the distribution grid and optimizes these groups separately to limit phase imbalances within the grid. In this case, we may use the approach presented below to place the problem back into a larger node. Although these transformations may be performed automatically, so far we have implemented these transformations only by hand.

- 3.2.5.4.1. *Combining* In many cases, the natural hierarchical organization of the energy system leads to nodes with only a few (e.g. 10) subproblems, which is too small for many large scale optimization methods to be effective. In these cases, we may lump these nodes into a single overarching node. For example, we may choose to group multiple transformers that serve only a few households (as is common in the US and in rural areas), and at a higher level group small towns that have only one or a few transformers. If the nodes have any internal objectives or constraints, these should be represented within the new node. Note that from the optimization methods that we describe later in this chapter, currently only the column generation method (Section 3.4) can directly integrate such combined configurations. The other methods that we describe have to be adapted to be able to incorporate such configurations as well. Note that household nodes are a logical candidate for combination, although there may be technical (e.g. communication cost and latency) and cultural (privacy) reasons to keep these nodes separate and local.

For pass-through nodes (Section 3.2.5.3), the combining operation is trivial. Due to the linear structure of these nodes, subgraphs that contain only pass-through nodes can be rearranged arbitrarily without changing the meaning of the problem,

as long as the resulting subgraph still has the same descendants (and remains a tree). The same holds for profile steering pass-through nodes (that we introduce in Section 3.5.3.2), although there the organization may influence the structure within the solution.

3.2.5.4.2. *Splitting* In other cases, the natural structure leads to node problems that are too large to consider directly, e.g. for computation or communication reasons. A further argument in favour of small groupings is reliability, provided that these groups can be controlled locally. A typical case that potentially leads to nodes with too many direct children is a VPP problem. Such a VPP may control e.g. thousands of (non-localized) devices. For such cases, we want to be able to split a node into smaller parts. However, splitting a node is a difficult problem, unless the node has a special internal structure. Note that the decoupled optimization methods that we present in this chapter also rely on such structure, namely that we can describe the relevant behaviour of a node by its electricity demand over time and its internal cost.

Pass-through nodes also represent a simple case for splitting, where the idea presented in the previous section on the reorganization of pass-through node subgraphs can be used to introduce new internal pass-through nodes in the hierarchical structure. Note that a single pass-through node is also a (small) subgraph, and that we may split the children of this node into an arbitrary number of pass-through groups below it. By this, we may for example transform a pass-through node that has 1000 children (devices) to a master pass-through node that has 20 children, where the child nodes of this master node are again pass-through nodes that control 50 children (devices). The same principle applies to the profile steering pass-through nodes. Note that such an internal pass-through structure is also present in many column generation problems (Section 3.4), for example where a set of child nodes contribute (only) to the same demand balance equation. In this case, we may substitute these children in this balance equation of the master problem with an auxiliary “child” that adds up the profiles of the child nodes at hand, and split off this auxiliary child. This auxiliary child may be represented with a pass-through node, as it only adds up the profiles from the children below it (see Section 3.4.4.3).

3.3. PRICE STEERING WITH IDDP

3.3.1. INTRODUCTION

In [40: Section 3.5], a DP based local search heuristic is introduced for the balancing of the output of a group of microCHPs, which is extended by [19] and [233] to be used in a more general smart grid context, and named *iterative distributed dynamic programming* (IDDP). IDDP provides a scalable method for the balancing of large groups of devices, and is the default planning method of TRIANA. Although the name implies that the lower subproblems are solved by DP, we do not have specific restrictions that require the use of DP at the lower levels. The lower subproblems $i \in \mathcal{I}_x$ of a node x only have to support optimization for a price vector c_i^t . The approach aims to produce prices c_i^t for each i such that the cost (mostly

determined by prices $\mathbf{c}_x^\ell$) is minimized, while the demand remains between some bounds. However, this mechanism can also be used to follow some goal profile by choosing suitable bounds, which is the most common way that IDDP is used.

In this section, we place IDDP in the framework of Section 3.2. IDDP is typically used at a neighbourhood level, with intermediate pass-through nodes below it at household level, and devices connected to the household level (as in Figure 3.3, where in this figure an additional pass-through node is given at a higher level).

We discuss IDDP as follows. First, we define the problem that IDDP addresses more precisely in Section 3.3.2, followed by a description of the algorithm in Section 3.3.3. The main operation of IDDP is the price update rule, which we discuss in Section 3.3.4. We perform some small experiments with IDDP in Section 3.3.5 to enable a comparison with the optimization methods that we develop later in this chapter. We end with an evaluation in Section 3.3.6.

3.3.2. PROBLEM STATEMENT

In principle, IDDP addresses a price optimization problem with a price vector $\mathbf{c}_x^\ell$:

$$\min_{\mathbf{x}_x, z_x^{\text{down}}} \tau \mathbf{c}_x^\ell \mathbf{x}_x + z_x^{\text{down}}, \quad (3.6)$$

where we want to keep the demand $\mathbf{x}_x$ of the considered group below x within certain bounds $\mathbf{P}_x^{\text{upper}}$ and $\mathbf{P}_x^{\text{lower}}$:

$$\mathbf{P}_x^{\text{lower}}(t) \leq \mathbf{x}_x(t) \leq \mathbf{P}_x^{\text{upper}}(t) \quad \forall t \in \mathcal{T}. \quad (3.7)$$

The bounds may e.g. represent the rating of a transformer in both directions (feed-in and demand). Remember that z_x^{down} represents the costs within the children themselves (Section 3.2.3.2.2). In this case, the group of devices may maximize profit, but may have to limit the consumption and production at times with low and high prices, respectively, to stay between the bounds. Note that (3.6) and (3.7) are in general conflicting goals, as the bounds may restrict the amount of electricity that we can buy when it is cheap (to avoid excessive stress on the infrastructure), unless ample capacity is available.

In [19] and [233], the bounds are tightened to follow a goal profile $\mathbf{P}_x^{\text{goal}}$, with e.g. $\mathbf{P}_x^{\text{upper}} = \mathbf{P}_x^{\text{goal}} + \delta_x$ and $\mathbf{P}_x^{\text{lower}} = \mathbf{P}_x^{\text{goal}} - \delta_x$, with $\delta_x = 0.1 \left| \frac{1}{n_t \tau} \sum_{t \in \mathcal{T}} P_x^{\text{goal}}(t) \cdot \tau \right|$, i.e. δ_x corresponds to 10% of the average demand of x .⁶ The value of $\mathbf{P}_x^{\text{goal}}$ may be determined externally (and provided as a nonlinear component of $\mathbf{c}_x$), or derived from the result achieved in the initial iteration of the search procedure by taking the average power value over the optimization horizon ($P_x^{\text{goal}}(t) = \frac{1}{n_t \tau} \sum_{t \in \mathcal{T}} x_x^{(0)}(t) \cdot \tau$, for all $t \in \mathcal{T}$).

6. Note that the average demand may be nonrepresentative for δ_x if we e.g. have substantial feed-in, which may lead to an average demand of 0 W. In this case, we may instead use the peak-to-peak distance or the *root mean square* (RMS) value as the reference.

For practical reasons, the bounds P_x^{upper} and P_x^{lower} are often interpreted as soft constraints. In this case, we assign high weights $w_{x,t}^{\text{upper}}$ and $w_{x,t}^{\text{lower}}$ to the violations:

$$\min \tau c_x^{\ell \top} \mathbf{x}_x + z_x^{\text{down}} + \sum_{t \in \mathcal{T}} \tau \cdot (w_x^{\text{upper}} \max(0, x_x(t) - P_x^{\text{upper}}(t)) + w_x^{\text{lower}} \max(0, P_x^{\text{lower}}(t) - x_x(t))). \quad (3.8)$$

3.3.3. ALGORITHM

The basic idea of the algorithm is that we want to minimize (3.6) with as little impact from (3.7) as possible. Therefore, we first minimize (3.6) without considering (3.7), and then make as little changes to the prices as possible to satisfy the constraints. Note that the algorithm may be seen as a realization of the Lagrangian relaxation method (see Section 2.3.1.3.1).

The algorithm works as follows. First, we transfer the imposed pricing scheme c_x^{ℓ} of a given node x to initial prices $c_i^{(0)}$ for all children $i \in \mathcal{I}_x$. Based on the responses of the children to these prices, we iteratively update the prices: in every iteration $k \in \{0, \dots, n_k\}$ we ask each of the children to optimize their demand profiles $\mathbf{x}_i^{(k)}$ and cost $z_i^{\text{sub},(k)}$ based on updated prices $c_i^{(k)}$. The aggregate demand is given by $\mathbf{x}_x^{(k)} = \sum_{i \in \mathcal{I}_x} \mathbf{x}_i^{(k)}$, and is used to identify the time intervals where the bounds (constraints) are violated: let $\mathcal{T}^{\text{upper},(k)}$ be the time intervals where the upper bound is violated, and $\mathcal{T}^{\text{lower},(k)}$ the time intervals where the lower bound is violated, i.e.:

$$\mathcal{T}_x^{\text{upper},(k)} = \{t \in \mathcal{T} \mid x_x^{(k)}(t) > P_x^{\text{upper}}(t)\} \quad (3.9)$$

$$\mathcal{T}_x^{\text{lower},(k)} = \{t \in \mathcal{T} \mid x_x^{(k)}(t) < P_x^{\text{lower}}(t)\}. \quad (3.10)$$

The goal of the iterative algorithm is to empty these sets with minimal changes to the prices. If $\mathcal{T}_x^{\text{upper},(k)}$ and $\mathcal{T}_x^{\text{lower},(k)}$ are both empty, then we have a feasible solution. In this case, we terminate the search and return the current solution pattern $(\mathbf{x}_x^{(k)}, z_x^{\text{sub},(k)})$, with $z_x^{\text{sub},(k)} = z_x^{\text{down},(k)}$ (see Section 3.2.3.4). This solution has the potential to be a good representative of the minimization problem (3.6), as we have tried to add as little “artificial” cost as necessary to satisfy the constraints. If the infeasibilities are not yet resolved after a specified number of iterations n_k (i.e. $\mathcal{T}_x^{\text{upper},(n_k)}$ or $\mathcal{T}_x^{\text{lower},(n_k)}$ is nonempty), then we return the solution from the iteration k with the smallest violation of constraints (3.7): $\arg \min_k \sum_{t \in \mathcal{T}^{\text{upper},(k)}} \tau (x_x^{(k)}(t) - P_x^{\text{upper}}(t)) + \sum_{t \in \mathcal{T}^{\text{lower},(k)}} \tau (P_x^{\text{lower}}(t) - x_x^{(k)}(t))$.

3.3.4. PRICE UPDATES

To empty the violation sets, the prices for the children $c_i^{(k+1)}$ should be increased in the time intervals $\mathcal{T}_x^{\text{upper},(k)}$, and decreased in $\mathcal{T}_x^{\text{lower},(k)}$.⁷ However, the progress to empty these sets may not be stable: a price increase in time interval $t \in \mathcal{T}_x^{\text{upper},(k)}$

7. In principle, we can also influence the demand in $\mathcal{T}_x^{\text{upper},(k)}$ and $\mathcal{T}_x^{\text{lower},(k)}$ by changing the prices outside of these intervals, but the relation between our actions and the demand is more clear if we address these sets directly.

may remove t from $\mathcal{T}_x^{\text{upper},(k+1)}$, but may also add it to $\mathcal{T}_x^{\text{lower},(k+1)}$ if the children respond too eagerly, and vice versa for price decreases. This instability is particularly common when the bounds are close together, which is e.g. the case if we use a goal profile (as is the case when we use IDDP for balancing). Also, a price increase in $t \in \mathcal{T}_x^{\text{upper},(k)}$ may push demand to other time intervals that may then become part of $\mathcal{T}_x^{\text{upper},(k+1)}$. Furthermore, we should aim to limit the changes to the prices, as we want the problem to remain a good representative of (3.6). Therefore, we should be careful in how the prices are updated. To this end, several price update rules have been developed.

3.3.4.1. MICROCHP UPDATE RULE

In [40], a specific price update rule is presented for microCHPs. For every household $i \in \mathcal{I}_x$ with a microCHP, we have a set $\mathcal{T}_i^{\text{on},(k)}$ that indicates the time intervals where the microCHP is turned on, and also a set $\mathcal{T}_i^{\text{off},(k)}$ that indicates when it is turned off ($= \mathcal{T} \setminus \mathcal{T}_i^{\text{on},(k)}$). We increase the prices in the time intervals $t \in \mathcal{T}_x^{\text{upper},(k)}$, but only if the microCHP is turned off in interval t (and thereby does not produce electricity) in that particular house i . Conversely, we decrease the prices in the time intervals $t \in \mathcal{T}_x^{\text{lower},(k)}$, but only if the microCHP is turned on in time interval t (and thereby produces electricity) in i . Note that we may use this information as an extra termination condition for IDDP: if no household is able to decrease the infeasibility, then we know that the problem is infeasible. To decrease the price, we multiply it by a constant factor α ($0 < \alpha < 1$, where α is usually close to 1), and to increase the price, we multiply it by $2 - \alpha$ (note that these operations are slightly asymmetric). More formally, this gives:

$$c_i^{(k+1)}(t) = c_i^{(k)}(t) \cdot \begin{cases} \alpha & \text{if } t \in \mathcal{T}_x^{\text{lower},(k)} \cap \mathcal{T}_i^{\text{on},(k)} \\ 2 - \alpha & \text{if } t \in \mathcal{T}_x^{\text{upper},(k)} \cap \mathcal{T}_i^{\text{off},(k)} \quad \forall t \in \mathcal{T}, k \in \{0, \dots, n_k - 1\} \\ 1 & \text{otherwise} \end{cases} \quad (3.11)$$

An advantage of this rule is that it does not try to reduce prices if we already have the minimum demand from the device at hand, preventing unnecessarily low prices which may make the search less effective in subsequent iterations. However, the rule is susceptible to instability if the upper and lower bound are close to each other. Note that the roles of α and $2 - \alpha$ should be reversed to give the appropriate direction of change in time intervals with negative prices, and has no influence on intervals with a price of 0.

We can adapt this approach to a decentralized implementation by distributing $\mathcal{T}_x^{\text{lower},(k)}$ and $\mathcal{T}_x^{\text{upper},(k)}$, as every house knows whether its own microCHP is planned to be turned on. In this case, the overconsumption and underconsumption sets may be represented by two n_t -bit vectors (that may be further compressed).

We may also generalize this method to other devices and to groups of devices. First, note that the generalization to households with only a single controllable on/off device is straightforward, whereby we should switch the roles between the “on” and “off” states if we have demand instead of supply. If we have multiple devices, or devices with a continuous operating range, then we may look at the minimum and maximum demand that a child i can have at every time interval: if i is already at the minimum demand (or maximum supply), then the price should not be increased, and if i is already at the maximum demand (or minimum supply), then the price should not be decreased.

3.3.4.2. RANDOMIZED UPDATE RULE

The most commonly used price update rule for IDDP is introduced in [19: p. 74]. This rule is specifically designed to avoid the instability condition that we mentioned before. The idea is that we limit the number of children who receive a price update for a given time interval, which should lead to smaller demand changes and thereby reduce the risk of overshoot. We control this number by randomly choosing values that depend on the amount by which the bound is violated.

The rule works as follows. Let $p_{\mathcal{X}}^{\text{up},(k+1)}(t)$ denote the probability that a child of $\mathcal{X}$ receives a price increase for time interval t in iteration $k + 1$, and let $p_{\mathcal{X}}^{\text{down},(k+1)}(t)$ denote the probability to receive a price decrease. We choose these change probabilities in proportion to the bound violation:

$$p_{\mathcal{X}}^{\text{up},(k+1)}(t) = \begin{cases} \min(\beta_{\mathcal{X},t}^{(k)}(x_{\mathcal{X}}^{(k)}(t) - p_{\mathcal{X}}^{\text{upper}}(t)), 1) & \text{if } t \in \mathcal{T}_{\mathcal{X}}^{\text{upper},(k)} \quad \forall t \in \mathcal{T}, \\ 0 & \text{otherwise} \end{cases} \quad k \in \{0, \dots, n_k - 1\} \quad (3.12)$$

$$p_{\mathcal{X}}^{\text{down},(k+1)}(t) = \begin{cases} \min(\beta_{\mathcal{X},t}^{(k)}(p_{\mathcal{X}}^{\text{lower}}(t) - x_{\mathcal{X}}^{(k)}(t)), 1) & \text{if } t \in \mathcal{T}_{\mathcal{X}}^{\text{lower},(k)} \quad \forall t \in \mathcal{T}, \\ 0 & \text{otherwise} \end{cases} \quad k \in \{0, \dots, n_k - 1\}, \quad (3.13)$$

where $\beta_{\mathcal{X},t}^{(k)}$ is a scaling factor, e.g.:

$$\beta_{\mathcal{X},t}^{(k)} = \frac{1}{|x_{\mathcal{X}}^{(k)}(t)|} \quad \forall t \in \mathcal{T}, k \in \{0, \dots, n_k - 1\}. \quad (3.14)$$

Note that this $\beta_{\mathcal{X},t}^{(k)}$ is inappropriate if a (near-)zero demand values are possible (in this case, we may revert e.g. to the RMS value of $\mathbf{x}_{\mathcal{X}}^{(k)}$). Furthermore, as $\mathcal{T}_{\mathcal{X}}^{\text{upper},(k)}$ and $\mathcal{T}_{\mathcal{X}}^{\text{lower},(k)}$ are mutually exclusive, $p_{\mathcal{X}}^{\text{up},(k+1)}(t)$ and $p_{\mathcal{X}}^{\text{down},(k+1)}(t)$ are also mutually exclusively nonzero (we exploit this in some implementations to compactly represent the probabilities by using positive values to represent $p_{\mathcal{X}}^{\text{up},(k)}(t)$ and negative values to represent $p_{\mathcal{X}}^{\text{down},(k)}(t)$). As a reference, Bakker [19: p. 74] defines a deterministic “uniform price vector” variant which takes $p_{\mathcal{X}}^{\text{up},(k+1)}(t) = 1$ for $t \in \mathcal{T}_{\mathcal{X}}^{\text{upper},(k)}$, $p_{\mathcal{X}}^{\text{down},(k+1)}(t) = 1$ for $t \in \mathcal{T}_{\mathcal{X}}^{\text{lower},(k)}$, and 0 otherwise.

Using the above probabilities for changing the prices, we update the pricing schemes of the children by:

$$c_i^{(k+1)}(t) = c_i^{(k)}(t) + \begin{cases} c_i^{\text{up},(k+1)}(t) & \text{if } \mathcal{U}(0,1) < p_x^{\text{up},(k+1)} \\ c_i^{\text{down},(k+1)}(t) & \text{if } \mathcal{U}(0,1) < p_x^{\text{down},(k+1)} \\ 0 & \text{otherwise} \end{cases} \quad \begin{matrix} \forall i \in \mathcal{I}_x, \\ k \in \{0, \dots, n_k - 1\}, \\ t \in \mathcal{T}, \end{matrix} \quad (3.15)$$

where $c_i^{\text{up},(k+1)}$ and $c_i^{\text{down},(k+1)}$ give the step size of price changes, e.g.:

$$c_i^{\text{up},(k+1)}(t) = |c_x^\ell(t)| \frac{(n_k - k)}{n_k} \quad \forall i \in \mathcal{I}_x, k \in \{0, \dots, n_k - 1\}, t \in \mathcal{T} \quad (3.16)$$

$$c_i^{\text{down},(k+1)}(t) = -c_i^{\text{up},(k+1)}(t) \quad \forall i \in \mathcal{I}_x, k \in \{0, \dots, n_k - 1\}, t \in \mathcal{T}. \quad (3.17)$$

The magnitude of price changes decreases over k to reflect that later iterations should introduce smaller demand changes. However, note that the price step size does not tell how large the demand step will be, and furthermore note that the prices c_x^ℓ should not be close to 0 and should be of comparable magnitude to give useful price vectors $c_i^{(k)}$. A multiplicative update rule as presented in Section 3.3.4.1 may be more appropriate. In a decentralized implementation, we may broadcast the vectors $\mathbf{p}_x^{\text{up},(k+1)}$ and $\mathbf{p}_x^{\text{down},(k+1)}$, and let each child determine the price update locally.

This randomized update rule helps to reduce the mismatch (violation of the bounds), and makes the convergence much faster and more likely. However, the rule also has some disadvantages. The method produces randomized prices at a household level. Consequently, we also obtain randomized behaviour at a household level. This randomness is aggravated by the use of a pass-through node at household level, which gives every device the same (randomized) price vector and thereby motivates simultaneous high demand (which may cause local voltage problems in the electricity grid [159]). Households with multiple large consuming devices (e.g. including an EV with a large maximum charging power) may thereby give rise to overshoot conditions at a neighbourhood level. A possible workaround may be to distribute the price change *probability* to individual devices within the households (and apply the price update locally for every device). Alternatively, Krist [198: pp. 49–50] proposes to introduce a small quadratic term in the cost functions of the large devices, or to schedule these (often also very flexible) devices separately at a central level. As the scaling factor (3.14) is essentially arbitrary, we still might change the prices of too few or too many children at the same time: if we change too few prices, then the progress is slow, and if we change too many prices, then the approach suffers from overshoot. In Appendix C.1.2, we propose a *conditional* price update variant that first optimistically proposes price changes, and next decides to keep the price change only for a subset of the children. A more structural solution would be to make it explicit in the local objective of the children that excessive demand profiles are undesirable.

FIGURE 3.7: Node configuration of IDDP in experiments.

3.3.5. EXPERIMENTS

Bosman [40: pp. 79–85], Bakker [19: pp. 117–135], and Molderink [231: pp. 116–132] have already performed numerous experiments with IDDP. For completeness within the context of this chapter, we illustrate some planning results from the Flex Street case that we consider more extensively in Section 3.6 (and is introduced in Appendix B). The case considers the scheduling of 400 households, and aims for a flat demand profile. In these experiments, we use the group configuration of Figure 3.7, with $n_k = 30$ planning iterations. We use the randomized price update rule (Section 3.3.4.2) by default.

In Figure 3.8a, we present results to indicate the convergence behaviour of the planning process with IDDP in a single planning session. The other subfigures correspond with the optimization methods that we consider later in this chapter. We look at the planning session at 12:00 on the first day of the simulation, with a time horizon of 24 h. Each line represents the profile in a single iteration, and the bold line gives the final demand profile. The first iteration ($k = 0$) gives a very high peak demand of 855 kW, which may be taken as a representative for an uncoordinated approach. The figure shows that in the first few iterations ($k = 1, 2, 3$) the search introduces large new peaks where the demand used to be very low (i.e. overshoot), which generally settles after 4–5 iterations. After this, the profile improves slowly. The steering bounds $\mathbf{P}_x^{\text{upper}}$ and $\mathbf{P}_x^{\text{lower}}$ are illustrated as well (10% above and below the average in $k = 0$). The figure also shows the profile after $k = 10$ iterations (marked 10), and the winning profile with the least violation of the bounds, which occurs at $k = 29$. We see that the winning profile at some times follows the edges of the steering bounds quite well. However, with the given resources, it is not feasible to keep the profile within these steering bounds at all times. The randomized pricing scheme leads to a quite noisy control over the demand, which leads to a 10–20 kW spread of demand values (peak-to-peak) at times where it is in principle possible to realize a flat demand profile.

Peeking forward to the results for the other optimization methods in Figure 3.8, we see that these other methods give better results in terms of peak reduction ($\min \max \mathbf{x}_x$, where x refers to the root node), although these are not spectacularly

FIGURE 3.8: Planned demand during iterations of planning session at 12:00 on the first day of Flex Street (lower peak demand in last iteration is better).

better, i.e. the additional peak reduction is less than 5% in this specific planning session. When we consider these results, we have to keep in mind (as we noted in Section 3.1) that these small savings add up at scale (i.e. thousands of neighbourhoods). Furthermore, the other methods give more flexibility to consider objectives other than peak shaving. Finally, based on Appendix B.6, we show in Section 3.6 that it is theoretically impossible to improve the result much further for the case at hand.

3.3.6. EVALUATION

IDDP is an efficient decentralized optimization approach, but is not always as effective as we want it to be. Part of the reason for this is that we have fairly limited freedom to specify optimization problems within the given structure: we do not specify the objective that we want, but instead use some representative that fits in the structure of the solution method that we use. Note that this holds in general, yet IDDP has limited “modelling” parameters, i.e. the upper/lower bounds and

prices. Although it is sometimes possible to modify the approach to accommodate new objectives, this is generally difficult. Also, the randomized pricing scheme leads to a quite noisy control over the demand, which may be undesirable in practice, and may lead to a high number of switching operations for devices, which decreases their lifetime. Therefore, we investigate more sophisticated optimization methods in the rest of this chapter.

3.3.6.1. RECOMMENDATIONS

- 3.3.6.1.1. *Regularization* In the above, the problem of solving (3.6) subject to (3.7) is posed as a problem of determining an appropriate price update rule. However, it is questionable whether this can solve the problem in general, i.e. whether a combination of prices c_i exists that leads to a minimization of (3.6). More generally, we may approach the problem of (3.6) as a problem where some regularization is needed. The work in [198: pp. 49–50] suggests that this can help to reduce the impact of overshoot from large consuming devices. If we look at literature on Lagrangian relaxation (Section 2.3.2.1.2), which is closely related to IDDP, then we see that the problem is augmented with a quadratic penalty term in the objective that penalizes changes from the previous solution.
- 3.3.6.1.2. *Other optimization methods* Although these methods from literature promise to solve the convergence shortcomings of IDDP, in the remainder of this chapter we choose to look further at other options to solve the coordination problem.

3.4. COLUMN GENERATION

3.4.1. INTRODUCTION

IDDP, the earlier presented optimization approach for TRIANA (see Section 3.3), gives a very scalable and computationally efficient coordination mechanism, but has a limited effectiveness in terms of peak shaving. Part of the reason for this is that we are in principle not able to express this peak shaving objective within IDDP: whereas we can perform peak shaving with a specified upper bound (e.g. a transformer rating), we also want to make the peak demand value as low as possible, which is a more difficult problem. Furthermore, the randomized prices may lead to “ugly” profiles, with rapid on/off switching and seemingly contradictory control actions. Therefore, in this section we investigate more sophisticated optimization methods, which have more expressive power and give better scheduling results. However, to be relevant for our application, the method also needs to have an acceptable computational cost. Based on the results from Bosman [40: Chapter 6], column generation promises to have these properties, although the computational cost may be quite high. In this section, we present a column generation approach that fits in the framework of Section 3.2. We propose several changes to the basic column generation approach that lead to a significant decrease in computational cost, making it practically useful within the context of TRIANA.

Column generation is based on *linear programming* (LP), which is one of the most widely used techniques for mathematical optimization. Many problems can be naturally expressed as LPs, and efficient (polynomial) methods exist to solve such problems. Bertsimas [29] provides a background on linear programming. Although the underlying solution approach of IDDP (Lagrangian relaxation) may in principle address an even wider range of problems (for example, Hoogsteen [159] modified IDDP to account for internal grid constraints), we find that modifying these problems to accommodate other objectives is time consuming and error prone in practice. In contrast, LPs are relatively straightforward to use, and we can solve a wide range of problems with a straightforward interface and a single optimization mechanism. The LP-based approach that we propose in this section inherits these advantages.

If we consider the more general class of *mixed integer (linear) programming* (MIP), even more problems can be expressed, although this comes at the cost of an increased computational cost. LP (and if necessary MIP) gives a promising, flexible framework that allows users to specify a wide range of optimization problems, and may thereby help to specify the problem at hand more directly, reducing the mismatch between the real objective of the user and the objective that the optimization model addresses.

In principle, we can use an LP or MIP model to optimize the planning of a complete subtree within the model of Section 3.2, provided that we have a (mixed integer) linear model of the underlying resources in this subtree and the upward cost. However, using this complete subtree optimization approach limits scalability, and imposes extra restrictions on the structure of the nodes within the subtree. Also, the computation times of a MIP in general grow exponentially with the number of integer variables. As a result, solving large MIPs to optimality is not practical. This means that even though it may be possible to *specify* large (and a large class of) optimization problems as MIPs (and to a lesser extent LPs), this approach does not imply that these problems can be solved in reasonable time. Therefore, MIP models need to be designed with care; it is in particular important that the number of integer variables is kept small. Note that a MIP can be approximated by an LP by relaxing the integrality constraints (i.e. the constraint of a binary variable $x \in \{0, 1\}$ in the MIP is replaced by the constraint $x \in [0, 1]$ in the LP relaxation). This LP is much easier to solve, but it depends on the problem at hand whether the relaxed solution is still a good representative of the original problem.

Although we may use the approach sketched above to optimize small or restricted subtrees within our model, we want to have a general approach to partition the subtree in the optimization procedure. The main goal of this partitioning is to facilitate a decentralized implementation of the optimization procedure. We also want to remain able to use tailored optimization algorithms for the device nodes, as they may have a complicated nonlinear internal structure. Note that an LP formulation typically implies that devices can be turned on partially (e.g. 70 % now and 30 % the next day), which is not feasible in many cases. For example, a (normal) washing machine should always be turned on completely, and should run until it has completed its program.

In literature, decomposition provides a framework to partition optimization problems. Decomposition schemes restate hard optimization problems as an equivalent set of easier connected optimization problems. Decomposition problems are studied in depth by both the operation research and control engineering communities [284]. This has resulted in numerous practical approaches with different performance characteristics and assumptions on the problem at hand. In the context of smart grids, several decomposition algorithms have been considered, in particular dual decomposition (see [178]).

We focus on the Dantzig–Wolfe decomposition, which is a popular decomposition approach for (very) large scale problems that can be expressed as an LP. We introduce this approach in more detail in Section 3.4.2

Recently, Dantzig–Wolfe decomposition has received some attention in the context of DSM in smart grids as an efficient method to solve large control problems, as an alternative to dual decomposition and a centralized optimization, e.g. [6, 225, 226, 303, 306]. However, the mentioned papers do not consider the integer aspect, which is relevant in many practical smart grid problems. For example, most household appliances can only be turned either off or on, and larger devices often have a minimum operating level (e.g. EV charging). Also, some large grid-level resources have switching behaviour (e.g. transformers with tap changers and grid reconfiguration switches). The closest approach that we have found in literature on the application of Dantzig–Wolfe for energy management problems with integer subproblems is the work of [204], which considers the optimization of the interruption of groups of water heaters. The integer behaviour in this paper comes from the coupled signal for the control of large groups of devices. Bosman [40: Section 3.7, Chapter 6] uses the same decomposition principle (although this is not made explicit), and does consider an integer variant of a comparable problem.

In this section, we consider the application of column generation more generally for the decentralized planning problem in the model of Section 3.2. In the context of this model, we introduce the concepts of column generation and its mathematical basis, Dantzig–Wolfe decomposition, in Section 3.4.2. After this, we discuss the column generation algorithm for our problem, including some efficiency modifications, in Section 3.4.3. We follow this up with experiments with the algorithm in Section 3.4.4, considering different grouping configurations and the impact of our modifications. We defer a larger experiment and comparison with the other methods to Section 3.6. In Section 3.4.5, we conclude our discussion of column generation.

3.4.2. DANTZIG–WOLFE DECOMPOSITION AND COLUMN GENERATION

3.4.2.1. INTRODUCTION

For large LP problems, Dantzig and Wolfe introduced a decomposition technique in [72] that can be used for a large class of LPs to separate a given problem into its block structure, which can then be re-composed by means of column generation. Although the method is called “decomposition”, the problem at hand has never been “composed” in the first place. Column generation is a well-known tech-

nique from the *operational research* (OR) domain that is used to solve very large LP problems. It has been demonstrated to be highly effective for many problems that involve limited coupling between entities. The problems that we consider in this chapter have such a limited coupling: in principle, only the overall electricity demand behaviour of other nodes is relevant. As the area of machine scheduling problems was one of the domains where column generation was successfully applied and DSM has similarities with machine scheduling, Bosman previously also investigated the use of column generation in a specific large scale smart grid unit commitment problem [40: Chapter 6].

3.4.2.2. APPROACH

The decomposition approach works as follows. First, the LP is split up into a master problem and a group of subproblems. The master problem is concerned with a set of connecting variables and associated constraints. A series of subproblems is concerned with their own set of local variables and constraints. Subsequently, the master problem uses the subproblems to explore new *columns* of the original solution space by sending *shadow prices* to its subproblems. A column gives a possible assignment of the variables that are shared between the master problem and a subproblem, and also represents a solution for the subproblem. The shadow prices give the estimated value of the shared variables (or actually the constraints), and may therefore be seen as Lagrange multipliers, where we determine these prices (multipliers) from the LP solution structure by considering the dual of the LP. The subproblems have to be optimized subject to these prices to produce new columns for the master problem. By this, we only generate the parts from the subproblems that we need to find the optimal solution of the master problem. This approach gives optimal results: when there are no longer improving columns available for the subproblems, the given solution is optimal for the original problem.

A difference to most other Lagrangian based methods is that we may (and often do) continue to use large parts of earlier produced solutions, and may infer the existence of other solutions from the known set of solutions.

3.4.2.3. STRUCTURE

The approach imposes several requirements on the structure of the problem. First, the problem should be linear. Second, the coefficient matrix must have a block-angular (∇) structure as depicted in Figure 3.9. The top rows of the coefficient matrix $[\bar{A}_0 \cdots \bar{A}_n]$ and $\bar{\mathbf{b}}$ represent the connecting constraints: $[\bar{A}_0 \cdots \bar{A}_n] \cdot [\mathbf{x}_0^\top \cdots \mathbf{x}_n^\top]^\top = \bar{\mathbf{b}}$. Note that we use a slightly different symbol $\mathbf{x}$ instead of $\mathbf{x}$ to indicate that these variables do not (completely) correspond. The submatrices $A_1, \dots, A_n$ with right-hand side values $\mathbf{b}_1, \dots, \mathbf{b}_n$ represent the corresponding subproblems, with variables $\mathbf{x}_1, \dots, \mathbf{x}_n$ and constraints $A_1\mathbf{x}_1 = \mathbf{b}_1, \dots, A_n\mathbf{x}_n = \mathbf{b}_n$. Instead of equality constraints, $\leq$ and $\geq$ constraints may be represented as well. All blank parts of the matrix are zero. Furthermore, the connecting constraint submatrices $\bar{A}_1, \dots, \bar{A}_n$ in general tend to be sparse, and we can keep the variables that are not part of these connecting constraints private to the subproblems. In the con-

$$\begin{array}{c}
 \min \left[\mathbf{c}_0^\top \quad \mathbf{c}_1^\top \quad \cdots \quad \mathbf{c}_{n-1}^\top \quad \mathbf{c}_n^\top \right] \cdot \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} \\
 \text{s.t.} \quad \begin{array}{c} \overline{A}_0 \quad \overline{A}_1 \quad \cdots \quad \overline{A}_{n-1} \quad \overline{A}_n \\ \begin{array}{c} A_1 \\ \vdots \\ A_{n-1} \\ A_n \end{array} \end{array} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{n-1} \\ \mathbf{x}_n \end{bmatrix} \leq \begin{bmatrix} \mathbf{b} \\ \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_{n-1} \\ \mathbf{b}_n \end{bmatrix} \\
 \text{(almost) independent subproblem}
 \end{array}$$

FIGURE 3.9: Dantzig–Wolfe block angular structure in linear programs [72].

text of Section 3.2, this means that a subset of variables in $\mathbf{x}_i$ (with $i \in \{1, \dots, n\}$) may contain the electricity consumption $\mathbf{x}_i$ of a node i (which is determined in A_i and incorporated in a connecting constraint with $\overline{A}_i$), and a different part of $\mathbf{x}_i$ may e.g. describe the SoC over time (which is private and used only within A_i). By this, large parts of the matrix are independent, except for a set of complicating, connecting constraints.

3.4.2.4. INTEGER SUBPROBLEMS

In practice, we can work around the requirement that the problem has to be linear. We may just “pretend” that we have a linear problem. However, in this case we can no longer guarantee that we find the optimal solution for the original nonlinear problem. This is in fact the most common application of column generation, as large LP problems can mostly already be addressed directly, and column generation generally makes it slower. A popular application that exploits the use of nonlinear subproblems is the cutting stock problem [131, 132], using a knapsack formulation to generate cutting patterns based on the shadow prices from the master problem (a cutting pattern gives a possible option to cut up a roll of paper into a number of pieces of paper of a standard size). In this case, the master problem can choose how many rolls to cut up with the given patterns, and use the subproblem to generate new cutting patterns, where each roll size corresponds with a subproblem.

Although for MIPs it is possible to determine an optimal solution by column generation, this may require a lot of computational effort [23, 332, 333], and also communication effort in a decentralized implementation. Since we need to avoid a larger computation and communication effort, we do not consider this direction further. The structure imposed by the second requirement (block structure) is quite common in optimization problems, and is present in the problems that we consider: devices, households, and neighbourhoods are mostly technically independent from each other, except from an “upward” infrastructure connection.

3.4.2.5. COLUMN GENERATION PROCEDURE

3.4.2.5.1. *Restricted master problem* Based on the given decomposition, the problem at hand is solved with a column generation procedure. The problem is separated in a restricted master problem which represents the connecting constraints, and a set of subproblems (one for each subblock $A_i \mathbf{x}_i = \mathbf{b}_i$). The first is referred to as a *restricted master problem*, because it initially represents only a small part of the problem. The restricted master problem is incrementally built up with *patterns* from the subproblems, until it has the same optimal solution as the original master problem.

The subproblems are represented in the restricted master problem by pattern sets $\mathcal{P}_i$ for every subproblem i . A pattern $p \in \mathcal{P}_i$ gives a value $\mathbf{x}_{i,p}$ that has been generated by the subproblem i and leads to a feasible assignment of $\mathbf{x}_i$ with respect to the local constraints of i . In a linear context this means that $A_i \mathbf{x}_{i,p} = \mathbf{b}_i$. Note that the patterns $\mathbf{x}_{i,p}$ are parameters in the master problem that have been generated by the subproblems (following the approach of Section 3.4.2.5.2, where $\mathbf{x}_{i,p}$ is variable in the subproblem). Every pattern $p \in \mathcal{P}_i$ gives a column (variable) $y_{i,p}$ in the restricted master problem, with the coefficients $\bar{\mathbf{b}}_{i,p} = \bar{A}_i \mathbf{x}_{i,p}$ in the connecting rows and a weight $z_{i,p} = \mathbf{c}_i^\top \mathbf{x}_{i,p}$. The value $y_{i,p}$ indicates the number of times that pattern p from i is used. Using a pattern multiple times makes sense in the cutting stock problem, but in the Dantzig-Wolfe column generation variant we are restricted to a convex combination of patterns for each subproblem, i.e. $y_{i,p} \in [0, 1]$ and $\sum_{p \in \mathcal{P}_i} y_{i,p} = 1$ (giving an extra coefficient 1 in the restricted master problem for $y_{i,p}$). The set of patterns is updated iteratively, and we use a superscript (k) to describe the value in iteration k . The restricted master problem now has the form:

$$\min_{\mathbf{x}_0^{(k)}, \mathbf{y}_1^{(k)}, \dots, \mathbf{y}_n^{(k)}} \mathbf{c}_0^\top \mathbf{x}_0^{(k)} + \sum_{i \in \{1, \dots, n\}} \sum_{p \in \mathcal{P}_i^{(k)}} y_{i,p}^{(k)} z_{i,p} \quad (3.18)$$

$$\text{s.t.} \quad \bar{A}_0 \mathbf{x}_0^{(k)} + \sum_{i \in \{1, \dots, n\}} \sum_{p \in \mathcal{P}_i^{(k)}} y_{i,p}^{(k)} \bar{\mathbf{b}}_{i,p} = \bar{\mathbf{b}} \quad (3.19)$$

$$\sum_{p \in \mathcal{P}_i^{(k)}} y_{i,p}^{(k)} = 1 \quad \forall i \in \{1, \dots, n\} \quad (3.20)$$

$$y_{i,p}^{(k)} \in [0, 1] \quad \forall i \in \{1, \dots, n\}. \quad (3.21)$$

Note that we do not have to choose a specific pattern: the master problem may make convex combinations of the patterns of i , exploiting the property that any convex combination of patterns from $\mathcal{P}_i$ leads to a feasible assignment of $\mathbf{x}_i$ for linearly constrained problems. That is, for an $\mathbf{x}_i$ -vector of length 3, if $\mathcal{P}_i = \{\langle 1, 0, 1 \rangle, \langle 0, 1, 0.5 \rangle\}$, then $\mathbf{x}_i = \langle 1, 0, 1 \rangle$ ($\mathbf{y}_i = \langle 1, 0 \rangle$), $\mathbf{x}_i = \langle 0, 1, 0.5 \rangle$ ($\mathbf{y}_i = \langle 0, 1 \rangle$), and $\mathbf{x}_i = \langle 0.5, 0.5, 0.75 \rangle$ ($\mathbf{y}_i = \langle 0.5, 0.5 \rangle$) are all valid solutions. From $\mathbf{y}_i^{(k)}$, we may recover the solution for the original master problem with:

$$\mathbf{x}_i^{(k)} = \sum_{p \in \mathcal{P}_i^{(k)}} y_{i,p}^{(k)} \mathbf{x}_{i,p} \quad \forall i \in \{1, \dots, n\}. \quad (3.22)$$

When we use integer type subproblems, the number of times that we apply a pattern has to be a non-negative integer. In the earlier mentioned cutting stock problem, this means that we are not allowed to consume only half a roll of paper and then choose not to buy the rest of the roll. In general this also holds for the problem that we consider, with the additional constraint that the total number of selected patterns is equal to 1, which also leads to $y_{i,p}^{(k)} \in \{0, 1\}$. We may now consider the master problem as a column selection problem, which selects one pattern for every $i \in \mathcal{I}_\kappa$.

3.4.2.5.2. Column generation subproblem To generate new column (patterns) for the restricted master problem, we optimize the subproblem based on the shadow prices $\pi^{(k)}$ of (3.19), which are based on the (dual) solution for the problem (using the previously generated patterns, i.e. $\mathcal{P}_i^{(k)}$ for $i \in \{1, \dots, n\}$). The shadow prices describe the marginal cost of the right-hand side of a constraint on the objective. In the case of an integer problem, the shadow prices are in principle defined only for the LP relaxation of the problem. We also use the shadow prices of the y -constraints (3.20), which represent the cost of the current assignment for i .

The subproblems generate the solution/pattern/column that should lead to the most improvement in the master problem by finding an assignment of $\mathbf{x}_i$ that minimizes the reduced cost δ_i :

$$\delta_i = \min (\mathbf{c}_i - \pi \bar{A}_i)^\top \mathbf{x}_i - \tilde{\pi}_i. \quad (3.23)$$

$$\text{s.t. } A_i \mathbf{x}_i = \mathbf{b}_i, \quad (3.24)$$

with the definitions

$$\pi = \pi^{(k)}((3.19)) \quad (3.25)$$

$$\tilde{\pi}_i = \pi^{(k)}((3.20), i) \quad (3.26)$$

to match the notation in [71: p. 771], where (3.26) takes the i^{th} element of $\pi^{(k)}((3.20))$. In a nonlinear problem, (3.24) may also represent the more general constraints of i . Note that we may interpret the term $\mathbf{c}_i - \pi \bar{A}_i$ in (3.23) as prices on $\mathbf{x}_i$, and $-\tilde{\pi}_i$ as a constant offset. After this, we may check whether this column has the potential to lead to an improvement in the master problem, which is given if the reduced cost satisfies:

$$\delta_i < 0. \quad (3.27)$$

If this test is satisfied, then we should add it to the restricted master problem. It is also common to first ask all subproblems for their reduced cost value, and then only add the column from the problem with the most negative reduced cost.

In principle, if A_i has a suitable structure (or can be modified to take this structure, as discussed in Section 3.2.5.4.2), then we may decompose the subproblem i as well and use column generation to solve this problem, leading to a nested column generation approach [134].

3.4.2.5.3. *Procedure* Summarized, the general form of the column generation procedure for a Dantzig–Wolfe decomposition is as follows:

- Generate initial feasible set of columns $\mathcal{P}_1^{(0)} = \{\mathbf{x}_{1,0}\}, \dots, \mathcal{P}_n^{(0)} = \{\mathbf{x}_{n,0}\}$
- While improving patterns exist:
 - Solve (relaxed) master problem for iteration k
 - Solve subproblems: $\delta_i = \min_{\mathbf{x}_i} (\mathbf{c}_i - \pi \bar{\mathbf{A}}_i)^\top \mathbf{x}_i - \bar{\pi}_i$ s.t. $\mathbf{A}_i \mathbf{x}_i = \mathbf{b}_i$, with $\pi = \pi^{(k)}$ ((3.19)) and $\bar{\pi}_i = \pi^{(k)}$ ((3.20), i)
 - * Add solution to pattern set if reduced cost $\delta_i < 0$ (in case of a minimization problem). Some variants only add the pattern with the most reduced cost, or admit a maximum number of patterns.
- Solve integer master problem (if integer variant)
- Translate master problem solution to original problem solution

3.4.2.5.4. *Column generation with integer problems* One of the possible challenges with the basic column generation procedure, when it is applied with nonlinear subproblems, is that the LP relaxation of the master problem may grow to become unrepresentative of the MIP problem. For example, if a subproblem offers two demand patterns $\langle 1, 1, 0, 0 \rangle$ and $\langle 0, 0, 1, 1 \rangle$, then the relaxation of the master problem believes that the demand pattern $\langle 0.5, 0.5, 0.5, 0.5 \rangle$ exists as well. From this, we may conclude that we have found a balanced demand pattern. The relaxation is used to determine the prices for the subproblems, which may therefore become unrepresentative as well. If we for example have a second problem with a pattern set $\{\langle 1, 1, 1, 1 \rangle\}$, then we want the second subproblem to generate *either* e.g. the pattern $\langle 0.5, 0.5, 1.5, 1.5 \rangle$ (if we choose the first pattern for the first subproblem) or the pattern $\langle 1.5, 1.5, 0.5, 0.5 \rangle$ (if we choose the second pattern for the first subproblem). However, the relaxation would e.g. give the price vector $\langle 10, 10, 10, 10 \rangle$, which presumably gives again the demand profile $\langle 1, 1, 1, 1 \rangle$. This suggests that we should branch on specific pattern assignments, e.g. by first assigning the first pattern to the first subproblem, and then determining new prices (in literature, a more structured approach is known as *branch-and-price* [23]). Note that extreme cases as presented here are relatively uncommon for the type of problems that we have, especially for larger problems (with hundreds of subproblems). Also, in the example at hand, the first column generation iteration of the second subproblem would have generated the matching column for the first subproblem.

In the following, we tailor this general column generation approach to the problem of Section 3.2. Furthermore, we contribute several efficiency improvements to make the approach practical as a control method.

3.4.3. ALGORITHM

In this section, we tailor the general column generation procedure of Section 3.4.2.5 to problems with the structure of Section 3.2.

3.4.3.1. DECOMPOSITION

The first step to a column generation approach is the decomposition of the original problem by Dantzig–Wolfe decomposition. This step is straightforward if we already have the structure from Section 3.2: the optimization problem of a node $\mathcal{x}$ consists of the problem within the node $\mathcal{x}$ itself as the master problem, and the problems of the children of $\mathcal{x}$ (including their subtrees) as the subproblems. Note that whereas we decompose the problem along the hierarchical structure of the problem, it is also possible to decompose the problem e.g. along a scenario dimension [244]. Furthermore, combinations of both may also be possible in many cases. The connecting constraints are given by the demand balance equation (3.2) of node $\mathcal{x}$, which adds the contribution of the subproblems to the demand of $\mathcal{x}$. This scheme may be applied hierarchically, i.e. every child subtree represents a new problem. To make this more concrete, the neighbourhood level may correspond with the master problem, and the household level problems with the subproblems (which on their turn may be a master problem that has the device problems as subproblems). The internal problem of node $\mathcal{x}$, including its imposed pricing scheme but excluding the contribution of its children, may be described as an LP or as a MIP that can be well approximated by its relaxation (in the latter case, the method presented in Section 3.4.3.6 no longer applies).

3.4.3.2. MASTER PROBLEM

We first describe the master problem, i.e. the problem that belongs to a parent node $\mathcal{x}$, which coordinates the subproblems (children) $\mathcal{I}_{\mathcal{x}}$. This description follows more or less directly from the node model in Section 3.2.3 and from the general Dantzig–Wolfe column generation formulation in Section 3.4.2.5.

3.4.3.2.1. Balance equations/demand The central variables in the master problem formulation are the demand variables $\mathbf{x}_{\mathcal{x}}^{(k)}$, which describe the demand of node $\mathcal{x}$ over time intervals $\mathcal{T}$ in iteration k (i.e. the imbalance of node $\mathcal{x}$ that has to be “solved” by its parent). This demand is determined by the internal demand $\mathbf{x}_{\mathcal{x}}^{\text{int},(k)}$ and the demand of the subproblems $\mathbf{x}_{\mathcal{x}}^{\text{down},(k)}$:

$$\mathbf{x}_{\mathcal{x}}^{(k)} = \mathbf{x}_{\mathcal{x}}^{\text{int},(k)} + \mathbf{x}_{\mathcal{x}}^{\text{down},(k)} \quad (3.28)$$

$$\mathbf{x}_{\mathcal{x}}^{\text{down},(k)} = \sum_{i \in \mathcal{I}_{\mathcal{x}}} \mathbf{x}_i^{(k)}. \quad (3.29)$$

Internal nodes usually have an internal demand of zero:

$$\mathbf{x}_{\mathcal{x}}^{\text{int},(k)} = \mathbf{0}. \quad (3.30)$$

To work with this approach, we have to determine the demand $\mathbf{x}_i^{(k)}$ for each $i \in \mathcal{I}_{\mathcal{x}}$.

3.4.3.2.2. *Demand patterns, pattern selection* The (known) possible demand options for every subproblem $i \in \mathcal{I}_\mathcal{N}$ are given by the patterns $\mathcal{P}_i^{(k)}$. These sets are updated iteratively. Remember that the idea of column generation is that we can generate the best demand option dynamically (the generation of possible columns/patterns is covered by Section 3.4.3.3).

Although we specify the problem as if we have a new master problem for every iteration k , in practice the master problem is only incrementally modified. A pattern $p \in \mathcal{P}_i^{(k)}$ specifies the demand $\mathbf{x}_{i,p}$ over $\mathcal{T}$ that subproblem i may contribute to $\mathbf{x}_\mathcal{N}^{\text{down},(k)}$ (and thereby to $\mathbf{x}_\mathcal{N}^{(k)}$). A pattern p also specifies the subtree cost $z_{i,p}^{\text{sub}}$ for the child (subproblem) i that corresponds to this demand pattern choice. The pattern sets $\mathcal{P}_i^{(1)}$ should be initialized with at least one feasible pattern that is generated by the subproblem i (we discuss this initialization problem in Section 3.4.3.5). For every subproblem, we have to select a single demand pattern out of the set of options $\mathcal{P}_i^{(k)}$. This choice is represented by indicator variables $y_{i,p}^{(k)}$, which take the value 1 if we select p for i , and 0 otherwise. These constraints are modelled as:

$$\sum_{p \in \mathcal{P}_i^{(k)}} y_{i,p}^{(k)} = 1 \quad \forall i \in \mathcal{I}_\mathcal{N} \quad (3.31)$$

$$y_{i,p}^{(k)} \in \{0, 1\} \quad \forall i \in \mathcal{I}_\mathcal{N}, p \in \mathcal{P}_i^{(k)}. \quad (3.32)$$

For mathematical (duality) reasons, we can not use (3.32) directly, and instead have to use its linear relaxation within the column generation procedure, which gives:

$$y_{i,p}^{(k)} \in [0, 1] \quad \forall i \in \mathcal{I}_\mathcal{N}, p \in \mathcal{P}_i^{(k)}. \quad (3.33)$$

Using these pattern selection indicator variables, the demand of every subproblem i is given by:

$$\mathbf{x}_i^{(k)} = \sum_{p \in \mathcal{P}_i^{(k)}} y_{i,p}^{(k)} \mathbf{x}_{i,p} \quad \forall i \in \mathcal{I}_\mathcal{N}. \quad (3.34)$$

Note that in a practical implementation, we may choose to embed (3.34) directly in (3.29) (and (3.29) in (3.28)), unless the objective assigns cost or constraints to specific parts of the demand (other than $\mathbf{x}_\mathcal{N}^{(k)}$, e.g. the peak demand of the first three children).

3.4.3.2.3. *Costs* The objective of $\mathcal{N}$ (implicitly) uses the demand vector $\mathbf{x}_\mathcal{N}^{(k)}$ to determine the cost resulting from the demand. We define the (restricted) master problem for a given iteration k and root node $\mathcal{N}$ as follows (including the constraints from Section 3.4.3.2.1 and Section 3.4.3.2.2):

$$\min z_\mathcal{N}^{(k)}, \quad (3.35)$$

with the cost $z_{\mathcal{N}}^{(k)}$ following a structure similar to $\mathbf{x}_{\mathcal{N}}^{(k)}$:

$$z_{\mathcal{N}}^{(k)} = z_{\mathcal{N}}^{\text{up},(k)} + z_{\mathcal{N}}^{\text{sub},(k)} \quad (3.36)$$

$$z_{\mathcal{N}}^{\text{sub},(k)} = z_{\mathcal{N}}^{\text{int},(k)} + z_{\mathcal{N}}^{\text{down},(k)} \quad (3.37)$$

$$z_{\mathcal{N}}^{\text{down},(k)} = \sum_{i \in \mathcal{I}_{\mathcal{N}}} z_i^{\text{sub},(k)} \quad (3.38)$$

$$z_i^{\text{sub},(k)} = \sum_{p \in \mathcal{P}_i^{(k)}} y_{i,p}^{(k)} z_{i,p}^{\text{sub}} \quad \forall i \in \mathcal{I}_{\mathcal{N}}. \quad (3.39)$$

In this formulation, $z_{\mathcal{N}}^{\text{up},(k)}$ gives the upward cost (from the parent of $\mathcal{N}$) and $z_{\mathcal{N}}^{\text{int},(k)}$ the internal cost of $\mathcal{N}$. Both $z_{\mathcal{N}}^{\text{up},(k)}$ and $z_{\mathcal{N}}^{\text{int},(k)}$ should be modelled with an LP formulation (usually with some auxiliary variables and constraints), or as a MIP that is well represented by its LP relaxation. We give an example of an LP objective model later in Section 3.4.4.1. Depending on the context, the user may model the objective of $\mathcal{N}$ either as upward cost (which may only refer to $\mathbf{x}_{\mathcal{N}}^{(k)}$), or as internal cost (which may also refer to the demand of specific children, or groups of children). In practice, these cost variables are not made explicit, and are added directly to the master problem with linear objective coefficients (i.e. variable $y_{i,p}^{(k)}$ has a weight of $z_{i,p}^{\text{sub}}$). We refer to the LP relaxation of the master problem as (3.35)_r, and to the integer problem as (3.35)_i.

3.4.3.2.4. Subproblem pricing In the following, we consider the column/pattern generation subproblems of $i \in \mathcal{I}_{\mathcal{N}}$, which determine candidate patterns for $\mathcal{P}_i^{(k+1)}$. To determine these candidates, the subproblems need to know in what direction the restricted master problem wants to develop. This direction is indicated by linear prices $\mathbf{c}_i^{\ell,(k)}$, which are derived from the current solution to (3.35)_r.

To determine these prices, we could in principle follow the more formal price generation method from Section 3.4.2.5.2. However, we can take a shortcut that exploits our matrix structure, and in particular the structure of the balance equations such as (3.34). A constraint that we express e.g. as $x_{\text{sum}} = a_1 x_1 + \dots + a_m x_m$ becomes $a_1 x_1 + \dots + a_m x_m - x_{\text{sum}} = 0$ in the LP. Therefore, we should take the negative value of the shadow price that belongs to this *constraint* to find the marginal cost of the *variable* x_{sum} . To make this more clear, if we would formulate an equivalent constraint $2a_1 x_1 + \dots + 2a_m x_m - 2x_{\text{sum}} = 2 \cdot 0$, then the shadow price (π) of the constraint becomes half as large, and we would have to multiply the shadow price of the constraint by -2 to find the marginal cost of the variable x_{sum} . When we apply this to (3.34), we obtain:

$$\mathbf{c}_i^{\ell,(k)} = -\frac{1}{\tau} \boldsymbol{\pi}^{(k)}((3.34)), \quad (3.40)$$

where the factor $\frac{1}{\tau}$ again accounts for the convention in the model of Section 3.2 to express prices independent of the time duration (this constant is added back in (3.41)), and the negative sign is caused by the modelling of (3.34) in the LP as

discussed before in Section 3.4.2 (i.e. $\mathbf{x}_i^{(k)}$ has a coefficient -1 in (3.34)). Note that in some simple cases we may also take the shadow prices from (3.29) for all $i \in \mathcal{I}_\mathcal{N}$ (if there are no other constraints relating to specific $\mathbf{x}_i^{(k)}$) or from (3.28) (if also the internal demand is constant). The prices $\mathbf{c}_i^{\ell,(k)}$ represent the marginal cost of changes to the demand vector in the master problem.

3.4.3.3. COLUMN GENERATION SUBPROBLEM

3.4.3.3.1. Basic subproblem The subproblems $i \in \mathcal{I}_\mathcal{N}$ of the master $\mathcal{N}$ each try to find improving patterns for the master problem, based on the prices $\mathbf{c}_i^{\ell,(k)}$ that are provided by the master problem. The goal of this is to make the “expected” change in cost in the master problem as negative as possible:

$$\min_p \tau \mathbf{c}_i^{\ell,(k)\top} (\mathbf{x}_{i,p} - \mathbf{x}_i^{(k)}) + (z_{i,p}^{\text{sub}} - z_i^{\text{sub},(k)}) \quad (3.41)$$

$$\text{s.t. (specific constraints of } i \text{ and its subtree),} \quad (3.42)$$

where $\mathbf{x}_{i,p}$ gives the demand vector and $z_{i,p}^{\text{sub}}$ the cost of the newly generated pattern p , and $\mathbf{x}_i^{(k)}$ and $z_i^{\text{sub},(k)}$ give the current (possibly fractional) pattern assignment in the master problem (3.35)_r. The constraints in (3.42) are subproblem (device) specific and are not relevant in this discussion. Note that the current relaxed solution of the master problem for subproblem i has a reduced cost of 0; if this solution corresponds with a feasible pattern, then (3.41) can not be positive. The reduced cost test (introduced in (3.27), Section 3.4.2.5) now is performed on (3.41), meaning that if this value is smaller than 0, then subproblem i should propose the generated pattern p that minimizes (3.41) as an improving pattern to the master problem (which adds it to $\mathcal{P}_i^{(k+1)}$). Note that this test may be inaccurate in an integer context, i.e. a rejected pattern could still lead to an improved solution in (3.35)_r. Also, the subproblem is allowed to propose other patterns that do pass the reduced cost test. This may help to save computation effort especially in the first iterations where the pricing scheme is not very representative yet.

3.4.3.3.2. Simplified subproblem As the part of (3.41) that depends on the current pattern assignment for subproblem i in iteration k is constant, we may simplify (3.41) by adding $\tau \mathbf{c}_i^{\ell,(k)\top} \mathbf{x}_i^{(k)} + z_i^{\text{sub},(k)}$ to this problem, which leads to the equivalent minimization problem (up to a constant):

$$z_i(\mathbf{c}_i^{\ell,(k)}) = \min_p \tau \mathbf{c}_i^{\ell,(k)\top} \mathbf{x}_{i,p} + z_{i,p}^{\text{sub}} \quad (3.43)$$

$$\text{s.t. (specific constraints of } i \text{ and its subtree).} \quad (3.44)$$

This corresponds to a typical node optimization problem from Section 3.2 with imposed linear prices $\mathbf{c}_i^{\ell,(k)}$, which we may solve by any suitable method. The reduced cost test should compensate for the constant difference, i.e. the local view of cost $z_i(\mathbf{c}_i^{\ell,(k)})$ should be smaller than $\tau \mathbf{c}_i^{\ell,(k)\top} \mathbf{x}_i^{(k)} + z_i^{\text{sub},(k)}$ to admit the pattern in the master problem.

3.4.3.3. *Decentralized implementation* In a decentralized implementation, we may broadcast the prices $\mathbf{c}_i^{\ell,(k)}$ if these are equal for all subproblems $i \in \mathcal{I}_\mathcal{X}$ (or equal for almost all i , and correct the differences). Before subproblem i sends the generated pattern itself to the master problem, we may first perform the reduced cost test by sending only the objective value of (3.43) to the master problem, which may then decide to request only the improving (or only the most promising) patterns from the subproblems. Alternatively, we can perform the reduced cost test locally. To this end, each i in principle needs to know the assignments of the master problem to $\mathbf{x}_i^{(k)}$ and $z_i^{\text{sub},(k)}$, or at least the associated cost $\tau \mathbf{c}_i^{\ell,(k)\top} \mathbf{x}_i^{(k)} + z_i^{\text{sub},(k)}$. If i knows which patterns the master problem has used to generate the prices (e.g. all patterns produced so far), then i may locally reconstruct this cost. The cost corresponds to:

$$\min_{y'_i} \sum_{p' \in \mathcal{P}_i^{(k)}} y'_{i,p'} (\tau \mathbf{c}_i^{\ell,(k)\top} \mathbf{x}_{i,p'} + z_{i,p'}^{\text{sub}}) \quad (3.45)$$

$$\sum_{p' \in \mathcal{P}_i^{(k)}} y'_{i,p'} = 1 \quad (3.46)$$

$$y'_{i,p'} \in [0, 1] \quad \forall p' \in \mathcal{P}_i^{(k)}. \quad (3.47)$$

3.4.3.4. COLUMN GENERATION PROCEDURE

We define a variant of the procedure in Section 3.4.2.5 to solve the problem of $\mathcal{X}$, using the master problem of Section 3.4.3.2 (which represents $\mathcal{X}$) and the subproblems of Section 3.4.3.3 (which represent the children $\mathcal{I}_\mathcal{X}$). This variant aims to address some of the shortcomings associated with having nonlinear subproblems. We do not try to solve the problem exhaustively (as e.g. in [23], which may lead to a large computation and communication effort), but instead aim for a practically useful near-optimal heuristic. Note that whereas a column generation approach for a linear problem may in principle support very large master problems with 10000s of subproblems, the nonlinear subproblems may restrict the size of problems that we may address in reasonable time in a single master problem. We aim to make the master problem computationally efficient enough such that it does not become a highly limiting factor on the size of the (individual) problems that may be addressed. Note that if the master problem becomes too large, we are often able to split it up into multiple smaller problems with the techniques from Section 3.2.5.4.2, although this may also lead to a combinatorial growth in iteration count as discussed in Section 3.2.5.2.

Our aim is to cope with the pattern integrality problems that we discussed in Section 3.4.2.5.4. One way to address these problems is to consider only a small pattern set, which gives the LP less opportunity to determine inappropriate combinations of patterns. The pruning of the pattern set may be seen as a form of irreversible branching (although the pruning is in fact reversible, we have not ex-

ploited this yet). We have developed two pruning methods, which we describe in the following. From those, only the second method is used for the results achieved in the remainder of this section.

3.4.3.4.1. LRU pruning method In [HT:3], a variant of column generation is presented that addresses the above problem, as well as the growth of the pattern selection problem, by aggressive pruning of the pattern set. This variant generates an integer solution in every iteration, and uses only the last two selected patterns (which may even coincide) and the pattern that was just generated by the subproblems, i.e. we prune the *least recently used* (LRU) patterns. The method solves both the LP relaxation (to obtain the shadow prices) and the MIP in every iteration. In this method, solving a MIP in every iteration makes the method slow, in spite of the aggressive pruning and a “desired demand pattern” modification to the objective. Although we have been able to solve problems with hundreds of subproblems with this formulation, it leads to long computation times for the master problem and thereby forms a major bottleneck in the system, even though we have used workstation class hardware, liberal tolerances, a relatively short optimization horizon (around 14 h) to achieve results in somewhat reasonable time (almost a day of computation time was needed for the year simulation case of Section 3.6 with a slightly different configuration). Note that in practice we expect to use the planning more often than in simulation, e.g. every 15 min instead of every 6 h, and expect to use less powerful hardware in the field. Furthermore, we have found this method to be impractical in nested configurations with repeated invocations of the procedure (see Section 3.2.5.2). From this, it is clear that the MIP variant of the problem should be used sparingly.

A drawback of the low number of patterns is that this may lead to a non-smooth view in the master problem, and thereby give a nonrepresentative pricing scheme. For example, if we have a problem with the demand pattern $\langle 10000, 10001, 10000, 10000 \rangle$ and a peak minimization objective with weight 1000, then we generate the price vector $\langle 0, 1000, 0, 0 \rangle$, which technically gives (the opposite of) the correct direction of change for the demand profile, but clearly misses the point of peak minimization. This is especially a problem in the first iterations of the column generation procedure, and we address this problem in more detail in Section 3.4.3.5.

3.4.3.4.2. Multi-round pruning method To keep the computational effort of the column generation method limited, we should solve the integer master problem less often than proposed in Section 3.4.3.4.1. We may choose to solve a single large master problem MIP after we have generated a useful set of e.g. 5 – 10 columns for each subproblem based on the relaxation of the restricted master problem, which often turns out to be computationally less expensive than the combined effort for the individual MIPs in the earlier approach. However, we still want to be able to reduce the (possible) mismatch between the integer problem and its relaxation.

To this end, we propose an “inner/outer” iteration scheme. The inner iterations generate columns using only the LP relaxation of the problem. The outer iterations produce an integer solution from the available column sets for the subproblems. At the end of an outer iteration, we prune all patterns, except for the patterns

Input: imposed pricing scheme $\mathbf{c}_x$

Output: pattern (set) for x which assigns one pattern p_i for $i \in \mathcal{I}_x$

```

1 · embed  $\mathbf{c}_x$  in master problem (3.35)
2 · initialize  $\mathcal{P}_i^{(1,1)}$  for all  $i \in \mathcal{I}_x$ 
3 · for  $k_1 \in \{1, \dots, n_{k_1, x, I}\}$ :
4 ·   for  $k_r \in \{1, \dots, n_{k_1, x, r, k_1}\}$ :
5 ·     solve (3.35)r
6 ·     for  $i \in \mathcal{I}_x$ : # in parallel
7 ·        $\mathbf{c}_i^{\ell, (k_1, k_r)} \leftarrow -\frac{1}{\tau} \boldsymbol{\pi}^{(k_1, k_r)}((3.34))$  # constraint  $\mathbf{x}_i = \dots$ 
8 ·        $\mathcal{P}_{i, \text{cand}}^{(k_1, k_r+1)} \leftarrow \text{solve } i \text{ with imposed prices } \mathbf{c}_i^{\ell, (k_1, k_r)}$ 
9 ·        $\mathcal{P}_i^{(k_1, k_r+1)} \leftarrow \mathcal{P}^{(k_1, k_r)} \cup$ 
           $\{p : \mathcal{P}_{i, \text{cand}}^{(k_1, k_r+1)} \mid \tau \mathbf{c}_i^{\ell, (k_1, k_r)^\top} \mathbf{x}_{i, p} + z_{i, p}^{\text{sub}} < \tau \mathbf{c}_i^{\ell, (k_1, k_r)^\top} \mathbf{x}_i^{(k_1, k_r)} + z_i^{\text{sub}, (k_1, k_r)}\}$ 
10 ·   for  $i \in \mathcal{I}_x$ :
11 ·      $\mathcal{P}_i^{(k_1)} \leftarrow \mathcal{P}_i^{(k_1, n_{k_1, x, r, k_1}+1)}$  # possible pre-MIP pruning
12 ·   solve (3.35)I
13 ·   for  $i \in \mathcal{I}_x$ :
14 ·      $\mathcal{P}_i^{(k_1+1, 1)} \leftarrow \{p : \mathcal{P}_i^{(k_1)} \mid y_{i, p}^{(k_1)} = 1\}$  # pruning
15 ·    $\mathcal{P}_x^{(k_1)} \leftarrow \{(\mathbf{x}_x^{(k_1)}, z_x^{\text{sub}, (k_1)})\}$ 
16 ·  $\mathcal{P}_x \leftarrow \bigcup_{k_1 \in \{1, \dots, n_{k_1, x, I}\}} \mathcal{P}_x^{(k_1)}$ 
17 · return  $\{\arg \min_{p \in \mathcal{P}_x} z_{x, p}(\mathbf{c}_x)\}$ 

```

ALGORITHM 3.2: Inner/outer column generation algorithm.

that are part of the produced integer solution. We index the outer iterations by $k_1 \in \{1, \dots, n_{k_1, x, I}\}$, and the inner iterations by k_r . We generally choose a different inner iteration count for different outer iterations k_1 , and we denote the inner iteration count in node x in outer iteration k_1 as n_{k_1, x, r, k_1} , with $k_r \in \{1, \dots, n_{k_1, x, r, k_1}\}$. We denote the sequence of inner iteration counts as $\mathbf{n}_{k_1, x, r} = \langle n_{k_1, x, r, 1}, \dots, n_{k_1, x, r, n_{k_1, x, r}} \rangle$, with $|\mathbf{n}_{k_1, x, r}| = n_{k_1, x, I}$. For example, $\mathbf{n}_{k_1, x, r} = \langle 10, 2 \rangle$ indicates that we use 10 inner iterations in the first outer iteration, and 2 inner iterations in the second outer iteration. In the earlier description, we used the superscript $^{(k)}$ to refer to a variable or parameter in a specific iteration. We modify this to a superscript $^{(k_1)}$ where we refer to an outer iteration, and to $^{(k_1, k_r)}$ where we refer to an inner iteration. If at some place it is not clear whether we refer to the outer or inner iteration, we may also use the subscript _I to refer to an outer (“integer”) value, and the subscript _r to refer to an inner (“relaxation”) value.

The procedure is presented in more detail in Algorithm 3.2. We first embed the upward cost in the problem (line 1), which usually means that we set the objective coefficients of $\mathbf{x}_x$ to $\tau \mathbf{c}_x$ if this pricing scheme is a simple price vector (if there are further linear internal costs for $\mathbf{x}_x$, then we should add these as well). Fur-

thermore, we determine an initial pattern set for every subproblem $i \in \mathcal{I}_\mathcal{N}$ (line 2), which usually contains a single pattern. We discuss the initialization problem in more detail in Section 3.4.3.5.

Line 3 starts the outer (integer) iteration process, which is repeated $n_{k, \mathcal{N}, 1}$ times and indexed by k_1 . Within this process, we start the inner (relaxation) iteration process (line 4), which is repeated $n_{k, \mathcal{N}, r, k_1}$ times and indexed by k_r . The number of inner loop iterations depends on the current iteration k_1 . In the inner loop, we first solve the LP relaxation of (3.35) (line 5). We solve this problem to obtain the shadow prices from the constraints (3.34) for $i \in \mathcal{I}_\mathcal{N}$. The heuristic that we present later in Section 3.4.3.5.2 also uses the demand vector $\mathbf{x}_\mathcal{N}^{(k_1, k_r)}$ from this relaxed solution.

Next, for every $i \in \mathcal{I}_\mathcal{N}$ (line 6), we determine linear prices $\mathbf{c}_i^{\ell, (k_1, k_r)}$ from the shadow prices (line 7), which give the marginal cost of the demand variables $\mathbf{x}_i$ in the restricted master problem. We multiply the shadow prices by $-\frac{1}{\tau}$ to obtain the subproblem price vectors. In the first few iterations, the heuristic of Section 3.4.3.5.2 modifies these prices to improve convergence.

Subsequently, we solve the subproblem i , imposing the prices $\mathbf{c}_i^{\ell, (k_1, k_r)}$, which produces a candidate pattern $\mathcal{P}_{i, \text{cand}}^{(k_1, k_r+1)}$ (line 8). Although $\mathcal{P}_{i, \text{cand}}^{(k_1, k_r+1)}$ is described as a set, we usually obtain only a single pattern. We only add the pattern(s) from this candidate set to $\mathcal{P}_i^{(k_1, k_r+1)}$ if they pass the reduced cost test, which tests (based on the marginal costs) whether the inclusion of the pattern in the restricted master problem can lead to an improvement in its LP solution (line 9). This comes down to checking whether (the direction of) the proposed pattern $\langle \mathbf{x}_{i, p}, z_{i, p}^{\text{sub}} \rangle$ is better than the current (possibly fractional) LP pattern assignment $\langle \mathbf{x}_i^{(k_1, k_r)}, z_i^{\text{sub}, (k_1, k_r)} \rangle$. Note that column generation approaches that accept only a single pattern with the most negative reduced cost out of all generated patterns from all subproblems are quite common as well, aiming to keep the restricted master problem as small as possible. However, in our case we try not just to limit the size of the restricted master problem, but also try to limit the amount of work (computation and communication) for the subproblems. If no subproblem can contribute a pattern with negative reduced cost, then we may terminate the inner loop. This concludes the inner loop.

During the inner loop, we have obtained pattern sets $\mathcal{P}_i^{(k_1, n_{k, \mathcal{N}, r, k_1+1})}$ for all subproblems $i \in \mathcal{I}_\mathcal{N}$, which give the patterns that we can choose from for every i . Before we solve the integer restricted master problem, we may first prune some of the available patterns to limit the size of this problem, for example based on the column selection weights $\mathbf{y}_i^{(k_1, n_{k, \mathcal{N}, r, k_1+1})}$ (note that we first have to solve problem (3.35)_r again to obtain these values). For now, we just accept all the patterns from $\mathcal{P}_i^{(k_1, n_{k, \mathcal{N}, r, k_1+1})}$ in the integer restricted master problem (line 11); we denote the integer restricted master problem pattern sets as $\mathcal{P}_i^{(k_1)}$ ($i \in \mathcal{I}_\mathcal{N}$). We subsequently solve the integer restricted master problem (line 12). The typical approach to solve this problem is by a MIP solve; however, this approach is slow in many cases, and we discuss alternatives in Section 3.4.3.6. The integer restricted master problem solution is

used to prune the subproblem pattern sets (line 14), and to generate a candidate pattern for $\mathcal{X}$ that is based on the selection of one pattern for every subproblem $i \in \mathcal{I}_{\mathcal{X}}$ (line 15). The pruning aims to remove the mismatch between the relaxation and the integer solution. We realize this by carrying over only the pattern that has been selected for i in the integer restricted master problem. The pattern generation follows straightforwardly from the integer restricted master problem solution. Although the generated pattern is stored in a set $\mathcal{P}_{\mathcal{X}}^{(k_i)}$, in practice we produce only a single pattern in every outer iteration. This concludes the outer loop.

During the outer loop, we have obtained pattern sets $\mathcal{P}_{\mathcal{X}}^{(k_i)}$ for $k_i \in \{1, \dots, n_{k_i, \mathcal{X}, I}\}$, which contain possible (feasible, presumed near-optimal) solutions for $\mathcal{X}$. We lump these solutions into a single set $\mathcal{P}_{\mathcal{X}}$ (line 16), and select the solution/pattern p from this set with the lowest local view of cost $z_{\mathcal{X}, p}(\mathbf{c}_{\mathcal{X}})$ (line 17). Note that we may in principle also pass all of these solution to the parent, and let the parent decide which pattern to use. This concludes the definition of the inner/outer column generation procedure.

We note that the LRU method from Section 3.4.3.4.1 can be seen as a variant of this method by taking $n_{k_i, \mathcal{X}, I}$ as the iteration count n_k , using only a single inner iteration $\mathbf{n}_{k, \mathcal{X}, I} = \langle 1, \dots, 1 \rangle$, and replacing $\mathcal{P}_i^{(k_i)}$ with $\bigcup_{k \in \{k_i, k_i-1\}} \mathcal{P}_i^{(k)}$ on line 14, which retains the patterns that have been selected for i in the last two integer restricted master problem iterations (giving a total of at most three patterns for every i in the integer restricted master problem if every subproblem contributes at most a single pattern at line 8).

3.4.3.5. INITIALIZATION AND CONVERGENCE

A disadvantage of a column generation procedure is that it may converge slowly in many cases, being first slow to start (“heading-in effect”), and later slow to converge to the optimal solution (“tailing-off effect”) [214]. Since an approximate solution is acceptable in most cases, mainly the first problem is relevant here. In the first iterations, the restricted master problem has little information on the structure of the problem, which may lead to extreme behaviour of the shadow prices, as demonstrated earlier (e.g. on p. 87). This problem can be addressed both by choosing good initial columns and by alternative pricing rules [214]. We discuss the used initialization method in Section 3.4.3.5.1. After this, we discuss an alternative pricing heuristic that we use in the first iterations in Section 3.4.3.5.2.

- 3.4.3.5.1. *Initial pattern set* To start a column generation algorithm, we first have to seed the restricted master problem with a set of columns that gives a feasible (relaxed) solution. One of the potential problems with the column generation method is that the restricted master problem can be infeasible, which leads to an unbounded dual solution, and therefore also to unbounded shadow prices. Note that in this context only the relaxation of the LP has to be feasible, and not the integer problem. For now, we choose to ignore the possibility that a pattern selection can lead to an infeasible solution, and assume that the auxiliary constraints that describe the objective only give soft constraints, and may introduce slack variables with very

high cost to represent “hard” constraints. Dantzig [72: pp. 772–773] gives a more complete initialization procedure that constructs a feasible initial solution for LP column generation problems. For integer problems, Lübbecke [214: Section 4.1.1] gives several references on the initialization of column generation for such problems, including [332: p. 69].

Although it is typically suggested to start with a *set* of columns for each subproblem, we choose to start with only a single pattern for each subproblem. We characterize this pattern by the pricing scheme $\mathbf{c}_i^{(0)}$ that is used to generate it. The reason for this is that the generation and transfer of these initial patterns also induces computation and communication effort, and we may initially not know in which direction this effort should be spent. We prefer spending 1 – 2 extra iterations to the effort to build up the initial pattern set.

The most straightforward choice for the initial costs $\mathbf{c}_i^{(0)}$ (and the choice that we make in most cases) is $\mathbf{c}_x^\ell$, i.e. just copy (the linear part of) the objective to the subproblems. In some cases this directly gives the optimal solution to the problem. This is the case if none of the internal constraints of $\mathcal{X}$ or the pricing model is active. We may also try to let each subproblem solve $\mathbf{c}_x$ completely, although the effectiveness of this approach may depend on the objective that we pose. For example, if we ask $\mathcal{X}$ to keep the aggregate demand of 400 households below 400 kW, then asking each household to keep their demand below 400 kW is not going to help (however, asking each of them to keep the demand below 1 kW or 2 kW may lead to a good initial guess). The results in Section 3.4.4.5 suggest that in some settings letting each subproblem solve $\mathbf{c}_x$ completely can give a good start solution.

When we consider a nested configuration (and the node at hand is not at the top of the optimization tree), then we use the previous integer solution, which was generated in a previous call of the optimization process for this subproblem with a different pricing scheme, as a start solution. This presumes that the given prices from the parent differ not too much from the previous pricing scheme.

If we have a balancing type of objective, then we believe that optimizing the subproblems against a quadratic objective may also be a good strategy to obtain the initial solution, based on the results later in Section 3.5.4. We have not tested this variant yet, but expect this to give good results.

3.4.3.5.2. *Bootstrap with alternative pricing* In the first iterations of the procedure, column generation may produce erratic prices for the subproblems, and it therefore is often worthwhile to use a pricing rule different from the usual simplex-based approach. We discuss a heuristic that significantly speeds up the initial convergence of balancing problems. Alternative pricing methods that aim to stabilize the procedure are discussed in a more formal setting in [214].

Bosman partially addresses this stability problem (in a problem specific way) by considering the number of microCHPs that are running in a certain time interval, which limits the “noise” from the start-up and shutdown effects of these devices [40: p. 99], which essentially means that the demand profile values are rounded.

Although this rounding heuristic may be generalized to a more general class of devices, the choice of the rounding step size gives a tradeoff between accuracy (which may lead to systematic over- or underestimations of the demand) and effectiveness in suppressing the noise. We discuss a more general approach to address these problems (stabilized column generation) in Section 3.4.5.1.1.

For the balancing problems that we often consider, we have developed a simple heuristic that gives good results, as we demonstrate in Section 3.4.4.3. The heuristic modifies the price vector $\mathbf{c}_i^{\ell, (k_i, k_r)}$ after line 7 in Algorithm 3.2, and relies on the idea that quadratic cost functions tend to give well balanced profiles. The heuristic in principle assumes that we want to minimize the distance from a zero demand profile, and emulates a quadratic cost on the aggregate demand vector $\mathbf{x}_{\mathcal{N}}^{(k_i, k_r)}$. To this end, we add a scaled version of the aggregate demand vector in the relaxation solution to the price vector. The scaled version of the aggregate demand vector represents a linear version of a balancing term. Where the original price generation method may be indifferent to where a maximum peak value is almost reached, the modified subproblem price generation method gives more information on where demand should be shifted to, which is relevant especially in the first iterations. The search should eventually switch over to the original price generation method (or a different method) to address the actual objective of $\mathcal{N}$.

The heuristic may be described as follows in the context of Section 3.4.3.4.2. We introduce a weight $B^{(k_i, k_r)}$ that brings the demand values to the same order of magnitude as the prices of the LP problem. This weight is gradually decreased during the iteration process. We add the following term after line 7 in Algorithm 3.2:

$$\mathbf{c}_i^{\ell, (k_i, k_r)} \leftarrow \mathbf{c}_i^{\ell, (k_i, k_r)} + B^{(k_i, k_r)} \mathbf{x}_{\mathcal{N}}^{(k_i, k_r)}. \quad (3.48)$$

We apply the heuristic only in the first $n_{k,B}$ iterations and only in the first outer iteration, i.e. for $k_i = 1$. We use the following weights $B^{(k_i, k_r)}$:

$$B^{(k_i, k_r)} = \frac{n_{k,B} - k_r + 1}{n_{k,B}} \cdot \frac{\max |\mathbf{c}_i^{\ell, (k_i, k_r)}|}{\max |\mathbf{x}_{\mathcal{N}}^{(k_i, k_r)}|}, \quad (3.49)$$

where $\mathbf{c}_i^{\ell, (k_i, k_r)}$ refers to the prices before (3.48) has been applied. The first half of the right-hand side of (3.49) gives a linear decay in k_r . The second half of the right-hand side of (3.49) adjusts the scaling of the prices to (the maximum value of) the demand profile. Note that we assume that the demand profile $\{0, \dots, 0\}$ does not occur. In previous work we assumed that e.g. $n_{k,B} = 2$ would be sufficient (as we have used in [BT:3]). However, the case in Section 3.4.4.3 asks for a larger number of warming up iterations. We now use $n_{k,B} = 20$, and use an adaptive limit to stop using the heuristic. We stop using the heuristic when the relative objective improvement between the inner iterations $\left(\frac{z_{\mathcal{N}}^{(k_i, k_r-1)}}{z_{\mathcal{N}}^{(k_i, k_r)}} \right) - 100\%$ falls below a certain threshold, for which we have chosen 1%.

When a modified price vector is used, we do not use the reduced cost condition to add a generated pattern and to conclude that the search is finished, because the condition may falsely conclude that improving patterns no longer exist.

3.4.3.6. NON-MIP COLUMN SELECTION

The reason that we still solve a MIP in the column generation procedure of Section 3.4.3.4 is that we have to decide on a single pattern to use for every subproblem. However, we note that we may also be able to select the patterns in a different way, and thereby avoid the computational cost of the MIP (and, in a commercial context, also avoid the licensing costs associated with state of the art MIP solvers). For example, we may replace the pattern selection in the MIP with a rounding based selection in its LP relaxation, i.e. we solve $(3.35)_r$ once more, and select the patterns for every $i \in \mathcal{I}_x$ with the highest weight. Note that this guarantees in no way that we obtain a good combination of patterns. However, we see later in Section 3.4.4.4.2 that the inner/outer iteration scheme can recover the situations where this column selection method leads to a poor solution. To improve the selection, we may use more sophisticated pattern selection methods, such as e.g. randomized rounding [271]. We find that the greedy algorithm of Appendix C.1.2 is effective as well. We may at this point also use different cost functions to assess the combined profiles, e.g. using a quadratic cost function instead of its piecewise representation in the MIP. Note that this objective only has influence on the selection of columns and not on the generation of columns for the subproblems.

We should note that this separate selection approach assumes that the master problem has no or limited decision freedom after the patterns are selected. If the master problem does have decision freedom apart from the pattern selection, then we still have to optimize the problem for a given pattern assignment. This is analogous to the separation between the unit commitment and the economic dispatch problem: we first determine which units (patterns) are committed, and then determine how these units (patterns) should be dispatched. The current definition of a pattern describes such flexibility within a node only implicitly.

In the following, we perform experiments with the presented column generation procedure to evaluate the performance of this method.

3.4.4. EXPERIMENTS

To illustrate the operation of the column generation algorithm, we perform some small experiments, followed by an evaluation of the convergence behaviour of the method according to the size of the groups that we plan. A more extensive evaluation follows in Section 3.6, where we consider a year simulation case in which we compare the column generation approach to the other optimization methods that we presented in this chapter.

3.4.4.1. MODELLING OF OBJECTIVE

One of the strengths of the column generation method is that we can straightforwardly model a wide variety of linear objectives. In the following, we define an example objective that we use in several experiments. We have also considered

other cost models, see e.g. the case in Appendix B.5. The experiments presented in Section 3.4.4.3, Section 3.4.4.4, and Section 3.4.4.5 use a slightly different, more complicated objective (as described in [M⁺T:3]).

We consider a peak shaving objective that is somewhat comparable to the one used within the IDDP approach. We omit the indexing of the variables by (k_i) and (k_i, k_r) . Although we could in principle embed the same objective as IDDP uses within the column generation problem (i.e. model the process of Section 3.3 within an LP), we choose an objective that is closer to our original intentions. For this, we consider a linear combination of multiple objectives.

The main objective is minimizing peak demand. To model this, we introduce a variable $x_{\mathcal{N},\max}$ that expresses the maximum of all $x_{\mathcal{N}}(t)$, with a corresponding objective weight $w_{\max}$:

$$x_{\mathcal{N},\max} \geq x_{\mathcal{N}}(t) \quad \forall t \in \mathcal{T}, \quad (3.50)$$

which is equivalent to $x_{\mathcal{N},\max} = \max_{t \in \mathcal{T}} x_{\mathcal{N}}(t)$ for positive $w_{\max}$. Next, we also want to avoid very low demand values (i.e. we want to avoid a very large variation in the demand values). We have chosen to express this as the minimization of the difference between the maximum and minimum demand value (i.e. the peak-to-peak distance). This difference is expressed by a variable $x_{\mathcal{N},\min}$ with objective weight $w_{\min}$:

$$x_{\mathcal{N},\min} \geq x_{\mathcal{N},\max} - x_{\mathcal{N}}(t) \quad \forall t \in \mathcal{T}, \quad (3.51)$$

which is equivalent to $x_{\mathcal{N},\min} = \max_{t \in \mathcal{T}} (x_{\mathcal{N},\max} - x_{\mathcal{N}}(t))$ for positive $w_{\min}$. Note that we could also have stated, using a variable $x_{\mathcal{N},\min}$ with a negative weight $w_{\min}$:

$$x_{\mathcal{N},\min} \leq x_{\mathcal{N}}(t) \quad \forall t \in \mathcal{T}, \quad (3.52)$$

which is equivalent to $x_{\mathcal{N},\min} = \min_{t \in \mathcal{T}} x_{\mathcal{N}}(t)$ for *negative* $w_{\min}$ (a lower minimum leads to a higher cost). These formulations are in fact equivalent, although (3.51) now contributes an extra weight $w_{\min}$ to $x_{\mathcal{N},\max}$.

When we optimized the problem with the above terms, we discovered that these do not completely express what we consider a good profile: with the above, we “obviously” do not intend to generate a solution that alternates between the lowest and the highest demand value, but this behaviour is not reflected in the objective and thereby not discouraged. Note that this is a problem with (the expectations of people from) optimization in general: we may intuitively expect a system to have some desirable behaviour that is expressed neither in the objective nor the constraints. Therefore, we have to add terms to the objective to make our intentions more clear. Alternatively, this could be solved by using a nonlinear (e.g. quadratic) cost function, or a piecewise linear approximation of such a (convex) cost function. A quadratic cost function (or in fact any strictly convex cost function) motivates more smooth optimal solutions. However, such nonlinear functions do not fit in the linear structure of the master problem, and the use of piecewise linear approximations may significantly increase the size of the master problem and thereby increase its computational cost.

To avoid unnecessary changes in the demand level, we add the following terms to the objective. We introduce variables $d_x(t)$ that express the absolute difference between consecutive values $x_x(t-1)$ and $x_x(t)$ with objective weight w_D :

$$d_x(t) = |x_x(t) - x_x(t-1)| \quad \forall t \in \mathcal{T}, \quad (3.53)$$

which is modelled in an LP by:

$$d_x(t) \geq x_x(t) - x_x(t-1) \quad \forall t \in \mathcal{T} \quad (3.54)$$

$$d_x(t) \geq x_x(t-1) - x_x(t) \quad \forall t \in \mathcal{T}. \quad (3.55)$$

The weight w_D must be positive. By this, we obtain solutions that are comparable in character to QP solutions with plateaus of stable demand (see e.g. Figure B.11a on p. 303), yet there is still no incentive to avoid lingering at a high demand level. Also, the solution may have abrupt changes in demand, which may lead to new problems (e.g. the synchronous start-up and shutdown of a large number of devices may give a synchronized demand peak at the edge of a time interval). We therefore use a variable $d_{x,\max}$ to limit the maximum value of the absolute differences $d_x(t)$, with objective weight $w_{D,\max}$:

$$d_{x,\max} \geq d_x(t) \quad \forall t \in \mathcal{T}, \quad (3.56)$$

which is equivalent to $d_{x,\max} = \max_{t \in \mathcal{T}} d_x(t)$ for positive $w_{D,\max}$. The above formulation still does not limit the frequency with which large demand change values occur: when $d_{x,\max}$ concedes to a high demand change somewhere on the relevant horizon, then this high value can also be used in many other intervals without any influence on the objective. We may somewhat limit this effect by restricting the intervals that are used to determine $d_{x,\max}$ to intervals in the first part of the planning period, i.e. we replace (3.56) by:

$$d_{x,\max} \geq d_x(t) \quad \forall t \in \{1, \dots, n_{td}\}, \quad (3.57)$$

where n_{td} is chosen such that it covers at least the time intervals up to the next planning session. This implies that we leave the minimization of the difference peaks in later time intervals to the planning sessions in the future. However, we note that here it might be more constructive to use nonlinear (strictly convex) costs (but do not act on this for now).

Finally, if we do not make explicit that buying electricity has a cost, then (3.51)/(3.52) may decide to dissipate electricity to raise the minimum demand value, for example by intentionally cycling the battery or by using an auxiliary resistance heating rod. We add linear commodity costs c_x to avoid these effects.

FIGURE 3.10: Node configuration of column generation in experiments.

Summarizing, we get the cost function:

$$\begin{aligned}
 z_{\mathcal{N}}^{\text{up}} &= \tau \mathbf{c}_{\mathcal{N}}^{\top} \mathbf{x}_{\mathcal{N}} \\
 &\quad + w_{\max} x_{\mathcal{N},\max} + w_{\text{mm}} x_{\mathcal{N},\text{mm}} \\
 &\quad + w_D \sum_{t'} \tau d_{\mathcal{N}}(t') + w_{D,\max} d_{\mathcal{N},\max}.
 \end{aligned} \tag{3.58}$$

$$z_{\mathcal{N}}^{\text{int}} = 0 \tag{3.59}$$

Note that although in this formulation we have chosen to represent the objective as upward cost, we may just as well represent the objective as internal cost, and set the upward cost to 0.

To reduce the amount of LP modelling effort for a specific case, we defined common operations in software that may be used orthogonally. For example, we may first determine the absolute difference variables of $\mathbf{x}_{\mathcal{N}}$, and then take the maximum of these variables to determine (and assign an objective weight to) $d_{\mathcal{N},\max}$.

3.4.4.2. EXAMPLE

Before we go into the more detailed experiments with the column generation method, we first give an example of the progress of planning during a column generation planning session. This example describes a single planning session at 12:00 with an optimization horizon of 24 h on the first day of the case that we evaluate in Section 3.6, i.e. the same case as we considered in Section 3.3.5. We use the group configuration of Figure 3.10, which is justified by the results of the experiments later in this section, and the objective of Section 3.4.4.1. The experiments use the rounding-based column selection (Section 3.4.3.6), and make use of the presented bootstrap price generation method (Section 3.4.3.5.2).

In Figure 3.8b (p.73), we present the development of the demand profile during the planning for column generation, together with the results for IDDP (Figure 3.8a) and profile steering (Figure 3.8c; this method will be introduced in Section 3.5). Although column generation is generally associated with a slow convergence, the bootstrap procedure helps to guide the search to a good solution. We find that the

search in every iteration roughly halves the relative distance between the profile peak demand and the final peak demand. However, note that the presented profiles represent the relaxed variant of the problem, and may thereby not represent a real available solution. At $k_1 = 1$, i.e. after $n_{k,x,r,1} = 10$ iterations, we obtain the first integer solution (grey, solid bold), and at $k_1 = 2$ (after $n_{k,x,r,2} = 2$ more iterations) we find the second and final integer solution (black, solid bold). In contrast to the LP solutions in later iterations, the integer solution has significant “noise” (about 5 – 10 kW peak-to-peak), in particular in the less important (non-peak) periods, although this is less extreme than in the IDDP solution. We expect that this may be improved using a more advanced column selection method; preliminary experiments show that the method from Appendix C.1.2 can decrease the distance between the LP solution and the integer solution. One of the explanations for the gap between the LP solution and the integer solution is that linear pricing asks for extreme points of the solution space, and these extreme points often use high demand values. Later in Section 3.5, we consider the use of nonlinear (quadratic) upward cost functions to obtain less extreme profiles.

3.4.4.3. CONVERGENCE BEHAVIOUR BY GROUP SIZE

3.4.4.3.1. Experiment In Section 3.4.3.4, we suggested to reduce the group size to make the master problem less difficult. To see the effect of this, we experiment with the convergence behaviour of the column generation procedure for different group sizes. We look at a single planning session from the Flex Street case as considered before. To have an interesting scenario without too much computational effort, we choose the start of the evening of the first day as the period of interest for the experiments (with a planning horizon from 12:00 to 2:15 on the next day). This time period gives sufficient time to avoid start-up problems, yet covers a relevant scheduling period: the demand profile must ramp up from a mid-day PV supply valley to the evening heat demand peak. To have an equal set up for all experiments, we always run the simulation up to the start of the chosen period with a baseline control method (using the node configuration of Figure 3.10, $n_{k,x,I} = 1$, and $\mathbf{n}_{k,x,r} = \{10\}$). Note that the experiments have been performed with a slightly more complex objective (as presented in [4]) and simulation configuration, although this should only have a negligible impact on the discussion at hand.

As pointed out in Section 3.2.5.2, a low iteration count is essential to make the problem scalable. Therefore, we evaluate the convergence behaviour. We run the algorithm until it terminates ($n_{k,x,I} = 1$ and $\mathbf{n}_{k,x,r} = \{\infty\}$). To evaluate the importance of group size, we make $|\mathcal{I}_x|$ groups of $400/|\mathcal{I}_x|$ houses. The corresponding group configuration is illustrated in Figure 3.11. For this experiment, we apply the method described in Section 3.2.5.3 at both the lower group level (at depth 1) and the house level (at depth 2). For the results, we expect that a smaller number for $|\mathcal{I}_x|$ makes the master MIP much easier to solve, but leads to the generation of more columns. The cases $|\mathcal{I}_x| \leq 10$ lead to similar results as for the optimization of the behaviour of an individual house, and the larger cases represent the optimization of a neighbourhood.

FIGURE 3.11: Nested node configuration with pass-through subproblems.

3.4.4.3.2. *Results* In Figure 3.12, we present the long term and short term convergence behaviour of the top level problem. We plot the objective value $z_{\mathcal{X}}^{(1,k_{x,t})}$ during the iterations of the relaxed master problem for various choices of $|\mathcal{I}_x|$. As can be seen in Figure 3.12a, the column generation procedure has a very long tail with marginal improvement. Therefore, we present the long term and the short term behaviour separately. The bootstrap method presented in Section 3.4.3.5.2 has a dominant effect on the short term convergence behaviour. To illustrate this effect clearly, Figure 3.12a focuses on the case without bootstrap, whereas Figure 3.12b focuses on the case with bootstrap. In each graph, semitransparent lines represent the other case.

Figure 3.12a demonstrates that, without bootstrap, a large number of subproblems $|\mathcal{I}_x|$ improves the convergence rate significantly. The column generation procedure can consider the smaller problems separately, which increases the flexibility of the master problem. For smaller $|\mathcal{I}_x|$, the master problem needs to request a new column to combine the columns from a lower level in a different way. Nevertheless, column generation manages to eventually find a good relaxed solution even when all houses are lumped into a single group. With the bootstrap procedure, the number of iterations to termination slightly decreases for large $|\mathcal{I}_x|$, and increases for small $|\mathcal{I}_x|$; a possible explanation for this is that this procedure initially does not follow the structure of the problem, which can be either good or bad.

For the short term convergence, the results (Figure 3.12b) look quite different. With bootstrap, the difference is much smaller. For all considered group sizes, the procedure already converges in 5 – 10 iterations. After 10 iterations, the objective difference between the smallest and the largest group size is only 6%. Regardless of group size, subsequent iterations offer only marginal improvement, whereby the largest extra improvement is found for $|\mathcal{I}_x| \geq 10$. To make this more clear, Figure 3.12c presents the progress at specific iterations in the process. Problems with a large $|\mathcal{I}_x|$ progress towards the final result far more quickly; for small $|\mathcal{I}_x|$, this process is very slow.

FIGURE 3.12: Convergence of column generation with subgroups, by group count $|I_N|$ and iteration count $n_{k,x,r,1}$.

FIGURE 3.13: Nested node configuration with pass-through subproblems.

FIGURE 3.14: Nested node configuration with pass-through group subproblems, multiple integer rounds.

3.4.4.4. INTEGRALITY COST

3.4.4.4.1. Group size In Section 3.4.4.3, we found that the *relaxation* of the column generation problem gives a good solution after $k_{\mathcal{X},r} = 10$ iterations. However, we do not yet know whether this also leads to a good solution if we also have to consider the *selection* of patterns (i.e. the integrality constraints): for every subproblem, we have to choose exactly one pattern, and not a linear combination of patterns. We expect that the integrality constraint has more impact for smaller groups, because it may be more difficult to find a good selection of patterns due to limited diversity between the households.

To investigate the relation between the group size and the gap between the relaxed solution and the integer solution, we compare these results for different values of $|I_{\mathcal{X}}|$. We use the group configuration of Figure 3.13. We present the objective value gap that results from the integrality constraint in Figure 3.15. For smaller $|I_{\mathcal{X}}|$, this gap is very large, and it becomes very small for larger $|I_{\mathcal{X}}|$. We see the fol-

 FIGURE 3.15: Integrality penalty by $|I_N|$, with MIP, $\mathbf{n}_{k,N,r} = \langle 10 \rangle$, and $n_{k,N,I} = 1$.

 FIGURE 3.16: Integrality penalty by column selection method (MIP, rounding) and number of top level children $|I_N|$.

lowing trend (also depicted in Figure 3.15 with a dashed line): the objective gap is almost equal to $\frac{1}{2|I_N|}$ for $|I_N| \geq 10$, and around $\frac{1}{|I_N|}$ for $|I_N| < 10$. A possible explanation for the difference is that for small $|I_N|$ the relaxed solution has not yet settled after 10 iterations.

During the experiments, we observed that the MIPs for the *smaller* problems actually take longer than those for the larger problems. This has to do with the way that we configured the solver (with a 1% MIP gap limit) and the structure of the problem.

- 3.4.4.4.2. *Integer mismatch removal* The LP relaxation may potentially diverge from the integer nature of the problem, and thereby no longer contribute any columns that are useful for the integer problem. In Section 3.4.3.4.2, we considered to eliminate the gap between the linear and the integer solution by performing a column selection, and then pruning all columns except for the selected columns. We experiment with this, using the group configuration of Figure 3.14. To avoid generating the same problem over again (i.e. such that the relaxation is no longer representative), we limit $n_{k,\mathcal{X},r,k_1}$ to 2 for $k_1 \geq 2$ ($\mathbf{n}_{k,\mathcal{X},r} = \langle 10, 2, \dots, 2 \rangle$).

We first use a MIP formulation to perform the column selection. The simulation results presented in Figure 3.16 show that this pruning method does not give any substantial improvement. We believe that it may be impossible to generate a suitable column from the subproblems without modifications to the problem, because linear pricing typically leads to extreme profiles, i.e. no price vector maps to a suitable profile when $|\mathcal{I}_{\mathcal{X}}|$ is too small.

In Section 3.4.3.6, we furthermore proposed to replace the MIP with a rounding based pattern selection. Figure 3.16 includes the results of this approach (labelled by the subscript _{round}). At $k_1 = 1$, the objective value is a lot worse than for MIP: even for large $|\mathcal{I}_{\mathcal{X}}|$, adding the highest-weight patterns together often gives a poor solution. However, at $k_1 = 2$, the method recovers the patterns it has pruned but which are in principle good for the overall integer solution, which almost fully eliminates the difference with the MIP version of the problem. This means that we can practically choose not to use the MIP altogether. Consequently, the problem becomes computationally much easier. This trades the effort on the master problem with extra subproblem effort. Note that it may also be possible to generate the columns from the patterns that were pruned earlier (i.e. by performing a reduced cost test on these patterns), however we have not investigated this.

3.4.4.5. NESTED COLUMN GENERATION

- 3.4.4.5.1. *Experiment* In a last series of experiments, we want to evaluate the behaviour of column generation with a cascaded problem configuration, as described in Section 3.2.5.2. We use the group configuration as illustrated in Figure 3.17. We partition the group of 400 houses in 20 groups of 20 houses ($|\mathcal{I}_{\mathcal{X}}| = 20$), and we take the same global objective as in Section 3.4.4.3. We also use this objective within each of the 20 groups, which means that we account for local peaks and demand changes in the network with the same weight. In this case, the local objective clearly supports the global objective: we expect that schedules that are good on a local scale are together also good globally. We control the number of iterations at the lower level ($n_{k,i,r,1}$ for $i \in \mathcal{I}_{\mathcal{X}}$), and take the number of iterations at the upper level ($n_{k,\mathcal{X},r,1}$) as a parameter (horizontal axis). In these experiments we choose not to prune the columns from the lower column generation problems, such that these may contribute in subsequent iterations of the upper column generation problem.

3.4 COLUMN GENERATION

FIGURE 3.17: Nested node configuration for experiment of Section 3.4.4.5.

3.4.4.5.2. Results In Figure 3.18, we present the results of these simulations. We break down the objective value by source: the upper curves correspond to the sum of the subproblem objective values ($z_{\mathcal{X},r}^{\text{down}}$), and the lower curves represent the objective value of $\mathcal{X}$ without considering the costs within the subproblems; we denote this cost as $z_{\mathcal{X},r}^{\text{top}} = z_{\mathcal{X},r} - z_{\mathcal{X},r}^{\text{down}}$. Due to reduced economy of scale at the subproblem level, $z_{\mathcal{X},r}^{\text{down}}$ is greater than $z_{\mathcal{X},r}^{\text{top}}$. Note that the column generation problems at the lower level contribute downward cost to the top level problem (which is part of $z_{\mathcal{X},r}$, but not of $z_{\mathcal{X},r}^{\text{top}}$). For reference, we also include the results of Section 3.4.4.3 corresponding to $z_{\mathcal{X},r}^{\text{top}}$ for $|\mathcal{I}_{\mathcal{X}}| = 20$ with a black thin line ("top level only"), which replaces the lower column generation problems with pass-through problems.

The results show that a bottom level iteration count of 1 ($\mathbf{n}_{k,i,r} = \langle 1 \rangle$) gives very slow convergence: the subproblems have little room to explore solutions that are acceptable for both the master and the local objective. For $\mathbf{n}_{k,i,r} = \langle 2 \rangle$ and higher values, the results are far better: the local objective already solves most of the global problems. However, note that the complexity scales exponentially by depth, as the iterations have to be performed in a nested way at each level (see Section 3.2.5.2).

Despite that the global and the local problems use the same objective, there are conflicts between the interests of the master problem and the subproblems. As a result, the values of $z_{\mathcal{X},r}^{\text{top}}$ and $z_{\mathcal{X},r}^{\text{down}}$ are not monotonically decreasing. To further point out this conflict, note that the value of $z_{\mathcal{X},r}^{\text{top}}$ in the found solution is 5% higher than for the solution achieved by the non-nested approach in Section 3.4.4.3 (i.e. the thin line "top level only" is slightly below the bundle of outcomes after $n_{k,\mathcal{X},r,l} = 10$).

3.4.4.5.3. Pre-optimization Looking at the results, we see that the problem spends a lot of the iterations on the top level to improve the local optimization of its subproblems. However, the system can also do this independently of the top level. Therefore,

FIGURE 3.18: Joint objective convergence of the top level problems and the subproblems by subproblem iteration count $n_{k,i,r,1}$.

it may be an option to “pre-optimize” the system locally with for example 20 column generation iterations ($\mathbf{n}_{k,i,r} = \langle 20 \rangle$) before doing the cascaded optimization with a low iteration count. Figure 3.18 includes the results for this with dashed lines (at the bottom of both line bundles). In this case, the local pre-optimization already almost solves the top level problem. Therefore for the following iterations a diverse column set is already available, and there is little need to generate new columns in the subproblems, which allows the use of a low iteration count value $n_{k,i,r,1}$. We think that it may even be possible to use only the columns that have already been generated (i.e. $n_{k,i,r,1} = 0$), provided that good columns are available for the problem at hand. By this, we can obtain a low iteration count, and thereby cope with the scalability problem of Section 3.2.5.2.

3.4.5. EVALUATION

As demonstrated in the previous sections, column generation is a powerful and flexible approach to solve very large linear optimization problems with a specific yet very common loosely coupled (Dantzig–Wolfe) structure between the different parts of the problem. As the optimization problems of Section 3.2 fit well with these characteristics, in particular at the neighbourhood level, we made an efficient realization of column generation for these problems. An important challenge here was

that our subproblems are in general not linear, and accommodating these problems in principle leads to a large and expensive MIP problem. However, we have shown that we may also address this problem with a fast rounding-based selection, and subsequently repair this solution. Also, by a suitable initialization and the reuse of previous solutions, we may shorten the iterative process. A local pre-optimization that accounts for the objective at higher levels helps further to reduce the number of nested iterations that we need to find suitable solutions.

Summarizing, column generation gives a scalable optimization approach for large groups of devices that gives improved results in comparison to the earlier scalable planning method IDDP, at a reasonable cost in computation and communication. As we demonstrate in Section 3.6, these results are near optimal in some cases.

The presented approach is less suitable for small problems (with few children, i.e. $|\mathcal{I}_N| < 20$), as the procedure may converge slowly in this case, and the integrality aspect becomes more relevant. We have also seen that the linear pricing of the subproblems leads to local extreme responses if there are no internal costs specified for high demand values at the lower levels, which may lead to local grid congestion if we can extrapolate the results for IDDP in [159]. Also, we have to make use of the flexibility of the approach (i.e. describe the objective more precisely and use modelling tricks) to produce useful solutions. Note that this is not a problem specific to column generation, but applies to LPs in general. In the current formulation, the presented approach is restricted to master problems formulated as an LP, and can therefore not (directly) describe other nonlinear constraints and objectives. The need for an LP or MIP solver is a practical barrier for the adoption of column generation. This applies in particular for state of the art MIP solvers, which have steep licensing costs. Also, in embedded environments there may be technical constraints that prevent the use of a solver (e.g. due to limited processing power, limited memory or nonstandard platforms).

3.4.5.1. RECOMMENDATIONS

- 3.4.5.1.1. *Better pricing heuristics* Although the presented pricing heuristic that we presented in Section 3.4.3.5.2 works well for balancing problems, we still see many improvement opportunities, in particular for other types of problems. We have a lot of problem specific information that is not yet exploited in the column generation procedure. In a rolling horizon scheme, we have not yet exploited the previous planning solution. The previous planning solution can contribute price information, as well as information on (the first part of) the possible demand profiles of the children. Also, many ideas from theory, as outlined e.g. in [214], have not yet been tried yet. Stabilized column generation seems to be one of the more promising ideas according to [214: Section 6.2.3]. In our context, this method may be interpreted as a battery within the restricted master problem (with high usage cost), which avoids too extreme price changes and which may gradually be made more expensive to use as the search progresses.

- 3.4.5.1.2. *Interior point column generation* A linear objective or at least the simplex method has some drawbacks, which we partly overcome with modelling tricks and special procedures. However, we would prefer to keep the problem more close to the basic problem formulation, and we also want to be able to describe at least quadratic terms (without a large piecewise expression). An interior point method based column generation approach [136, 137, 237] may address these drawbacks. This approach supports general large-scale convex problems (with Dantzig–Wolfe structure), and avoids the starting problems associated with a simplex-based approach by producing well-centered solutions. Although the integrality related problems remain, the work in [238] shows that the method may be used to produce integer solutions.
- 3.4.5.1.3. *Request the desired pattern* In the presented approach, we have to make a hard selection from the pattern set that is produced by the children of $\mathcal{X}$. This leads to a gap between the representation of a child within the column generation LP and the solutions that we may choose from (none of the columns is equal to the linear combination of columns in the current LP solution), which is relevant in particular if we have a small number of subproblems. To reduce this gap, we may specifically ask the subproblems to produce the patterns that are inferred by the LP (in the spirit of, and possibly using the procedure of Section 3.5). Also, we may try to solve the subproblems directly in the context of the current master problem solution to produce patterns (this is similar to the approach in Section 3.5.3). Whereas Section 3.4.5.1.1 proposes to stabilize the pricing (i.e. the dual solution), we here propose to stabilize the solutions that are produced within the subproblems. This idea is comparable to proximal methods such as ADMM [258], which rely on gradual updates of an existing solution to move towards the optimum (we do not know if this is a suitable strategy for mixed integer problems, as these are not always able to take gradual steps).
- 3.4.5.1.4. *More explicit flexibility* We may also address the integrality problem by expressing more freedom within the patterns, i.e. by dealing with the highly flexible demand within a household separately. In this way, the master problem can plan a part of the demand without generating new patterns, which makes it easier to resolve integer problems. This may in principle also be described as a type of columns. We may also represent the available flexibility with an energy flexibility space (Section 2.3.2.1.4), although this may overestimate the capabilities of the devices and underestimate their cost.
- 3.4.5.1.5. *Gradual integer commitment* Indecisiveness is one of the main challenges of the column generation approach relating to integrality, which may lead to making combinations of columns that do not lead to a feasible integer solution. A possible solution to this may be to fix the columns for a subset of the subproblems (e.g. half of the subproblems), and then continue the search with the rest of the subproblems. After this, the roles may be changed (i.e. one half is fixed, and the other half can be changed). We may increasingly fix values in this search, until we have found an appropriate solution. This is comparable to the branching in integer problems, although we consider a more randomized approach to fix columns.

3.5. PROFILE STEERING

3.5.1. INTRODUCTION

Many DSM approaches, including IDDP and column generation, use electricity prices as steering signals. For example, in its most common application, the approach in Section 3.3 tries to find a set of prices such that the aggregate demand profile of a group of devices follows a goal profile (e.g. is flattened). However, linear prices motivate extreme behaviour: one should consume as little as possible when prices are high, and as much as possible when prices are low (if the total demand is fixed). Therefore, it is generally hard, and in many cases even impossible, to find prices that result in e.g. a flat demand profile.

As an alternative, Gerards [HT:6] proposes an approach called *profile steering*: instead of using prices to express the desired behaviour of a node, we ask for the desired demand profile directly. The desired profile enters the objective of a node by penalizing the distance to this profile using e.g. a 2-norm, and may be part either of the internal cost (if the balancing is motivated internally) or the upward cost (if it is imposed by the parent node). In turn, a node may request a desired profile from its children to improve its own demand profile. This implies that the children have to support a profile-based pricing scheme.

Optimizing this class of problems is in general NP-hard, as shown in [HT:6]. Therefore, we propose a greedy heuristic to optimize this problem at a group level. We ask every child i to solve its problem in the context of the problem of its parent node x . Subsequently, the parent x selects the solution of the child that gives the largest improvement to its objective. We repeat this procedure until no child can suggest an improvement, or an iteration limit is reached. This is clearly a local optimization approach, which nevertheless proves to be very effective, in particular for nodes with a small number of children (e.g. a household) and devices with a non-binary control range (e.g. an EV with continuous or a large number of charging levels), for which IDDP is less suitable. An on-line interactive demonstration of this algorithm is available at [128].

As an example of applying profile steering, we consider the balancing problem of Figure 3.1 that was introduced on p. 43 as a base problem, and extend this example as shown in Figure 3.19. We express the goal of flattening the demand profile by giving x a desired demand profile of 0 W over the entire day (note that we may also choose any other fixed demand value here). Next, we ask every child to propose an initial solution, for example by requesting a flat profile of 0 W. This gives the demand profiles in the lower plots of Figure 3.1 that are marked o , and add up to the aggregate demand profile o in the upper plot. To reduce the distance of this aggregate demand profile to the desired profile, we request each child to realize a *difference* in demand that is equal to the difference between the aggregate demand profile and the desired profile (we denote this difference as d_x). In Figure 3.19, we illustrate the request profiles in the upper and lower plots as dotted lines (marked req.), together with the o -profiles and the corresponding response profiles (marked 1, 2 and 3 for each device, respectively). In the given example, the o -profile of each device has an RMS distance

FIGURE 3.19: Requested difference profiles and responses corresponding to Figure 3.1.

to the request profile of 2363 W ($\sqrt{\frac{1}{|d_x|} d_x^\top d_x}$). The profiles that are proposed by device 1 and 2 both have an RMS distance to the request profile of 2062 W and thereby give an improvement of 301 W (note that this distance corresponds to the distance of the aggregate profile after the proposed change is accepted). Device 3 proposes a profile with an RMS distance of 2121 W and thereby gives an improvement of 242 W. We now select either the improvement proposed by device 1 or device 2 (but not both). After this, we repeat the procedure, finding an improvement of 213 W for device 3, which is the last possible improvement. Note that although in this example we have found the optimal solution, this is not guaranteed in general.

In the following, we discuss profile steering in more detail. In Section 3.5.2, we describe the problem more formally. This is followed by a description of the algorithm in Section 3.5.3. We perform some small experiment results in Section 3.5.4, and evaluate the approach in Section 3.5.5.

3.5.2. PROBLEM STATEMENT

3.5.2.1. 2-NORM FORMULATION

In the following, we give a more formal description of profile steering. Let $\mathbf{p}_{\mathcal{X}}^{\text{goal}}$ be the goal demand profile for a node $\mathcal{X}$. The distance to this goal profile is incorporated in the objective with a weight $w_{p^{\text{goal}},\mathcal{X}}$. Both $\mathbf{p}_{\mathcal{X}}^{\text{goal}}$ and $w_{p^{\text{goal}},\mathcal{X}}$ are part of the imposed pricing scheme or introduced internally. This leads to the following objective:

$$\min w_{p^{\text{goal}},\mathcal{X}} \cdot \left\| \mathbf{x}_{\mathcal{X}} - \mathbf{p}_{\mathcal{X}}^{\text{goal}} \right\|_2 + z_{\mathcal{X}}^{\text{down}}, \quad (3.60)$$

where $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\top \mathbf{x}}$ is the 2-norm of $\mathbf{x}$. Note that the node $\mathcal{X}$ itself does not introduce extra constraints in this problem. To simplify the notation, we introduce a difference vector $\mathbf{d}_{\mathcal{X}} = \mathbf{x}_{\mathcal{X}} - \mathbf{p}_{\mathcal{X}}^{\text{goal}}$.

In principle, this definition is sufficient to describe the problem that we have in a node $\mathcal{X}$, and we can use the algorithm in Section 3.5.3 to solve this problem using the subproblems. However, we should still choose suitable weights for the variables, which we discuss in Section 3.5.2.2. Also, it is often more practical to express the problem as a quadratic problem. We give the quadratic formulation of the problem in Section 3.5.2.3

3.5.2.2. WEIGHTING

The objective proposed in (3.60) in its current form has some shortcomings if we aim to relate it to other costs. In particular, to support a nonuniform time interval length, we should weigh the demand values by the time interval length τ :

$$\min w_{p^{\text{goal}},\mathcal{X}} \cdot \left\| \sqrt{\tau} \mathbf{d}_{\mathcal{X}} \right\|_2 + z_{\mathcal{X}}^{\text{down}}. \quad (3.61)$$

Note that $(\sqrt{\tau} x)^2$ becomes τx^2 , which gives the intended weight for each demand value. By incorporating a constant weight factor in the objective, we may interpret the distance term as an RMS power value:

$$\min w_{p^{\text{goal}},\mathcal{X}} \cdot \sqrt{\frac{1}{\sum_{t \in \mathcal{T}} \tau}} \cdot \left\| \sqrt{\tau} \mathbf{d}_{\mathcal{X}} \right\|_2 + z_{\mathcal{X}}^{\text{down}}. \quad (3.62)$$

If the time interval length is uniform, then (3.62) simplifies to:

$$\min w_{p^{\text{goal}},\mathcal{X}} \cdot \sqrt{\frac{1}{n_t}} \cdot \left\| \mathbf{d}_{\mathcal{X}} \right\|_2 + z_{\mathcal{X}}^{\text{down}}. \quad (3.63)$$

Note that the objectives (3.60)–(3.63) have the same structure, and only differ in their weights on $\mathbf{d}_{\mathcal{X}}$.

A norm objective always considers some period, e.g. a day or a month, or a rolling window, similar to how one may account e.g. for the maximum value over some period. However, if the optimization horizon does not coincide with the accounting period (i.e. the period over which the cost is calculated), then we should

include a term to account for the RMS value outside of the horizon. This means that we consider the problem for the complete period, but can only influence the decisions on a short horizon. The part of $\mathbf{d}_x$ that is outside of the optimization horizon (but within the accounting horizon) may be represented with a constant within the balancing term. To determine this constant, we may use the known history of $\mathbf{d}_x$ for the window before the optimization horizon (the past), and an estimate of $\mathbf{d}_x$ for the window after the optimization horizon (the far future). Note that we should *not* try to spread the norm cost over the accounting window: for example, if we pay for the maximum demand over a month with an optimization horizon of a day, then we should not minimize the maximum demand on every individual day with a weight of $1/30$, but should on each day consider the full cost over the whole period (but may carry over the maximum value from earlier days, and estimate the maximum within the rest of the month).

3.5.2.3. QUADRATIC FORMULATION

As square root terms are difficult to integrate in optimization problems, we use a quadratic problem to represent (3.62). However, we have to keep the weight between the distance part and z_x^{down} consistent. If z_x^{down} is constant, then this problem does not occur: we may drop the constant $w_{\text{pgoal},x} \cdot \sqrt{\frac{1}{\sum_{t \in \mathcal{T}} \tau}}$, and as $\sqrt{\tau \mathbf{d}_x^\top \mathbf{d}_x}$ is monotonically increasing in $\tau \mathbf{d}_x^\top \mathbf{d}_x$, we may minimize the latter instead to find an optimal solution to (3.62).

If z_x^{down} is not constant, then we usually address (3.62) with an approximation:

$$\min w_{\text{pgoal},x} \cdot \sqrt{\frac{1}{\sum_{t \in \mathcal{T}} \tau}} \cdot \alpha_x \cdot \tau \mathbf{d}_x^\top \mathbf{d}_x + z_x^{\text{down}}, \quad (3.64)$$

where α_x approximates the effect of the square root operation. A logical choice for α_x is:

$$\alpha_x = \frac{\|\sqrt{\tau} \mathbf{d}_x\|_2}{\tau \mathbf{d}_x^\top \mathbf{d}_x} = \frac{\sqrt{\tau \mathbf{d}_x^\top \mathbf{d}_x}}{\tau \mathbf{d}_x^\top \mathbf{d}_x} = \frac{1}{\sqrt{\tau \mathbf{d}_x^\top \mathbf{d}_x}} = \frac{1}{\|\sqrt{\tau} \mathbf{d}_x\|_2}, \quad (3.65)$$

which may be evaluated in advance for a “typical” value of $\mathbf{d}_x$ or determined iteratively. Note that filling in (3.65) in (3.64) leads to (3.62).

Instead of using (3.64), we may also choose to specify the quadratic weight directly; in this case, the weight expresses a cost rate (e.g. in $\text{€}/(\text{kW})^2\text{h}$). This cost may e.g. represent the cost of the losses in the grid, which are quadratic in nature. More generally, we can consider a problem with quadratic weights $\mathbf{c}_x^q$ and linear weights $\mathbf{c}^\ell$, leading to the objective:

$$\min \tau (\text{diag}(\mathbf{c}_x^q) \mathbf{x}_x)^\top \mathbf{x}_x + \tau \mathbf{c}_x^\ell \mathbf{x}_x + z_x^{\text{down}}, \quad (3.66)$$

where $\text{diag}(\mathbf{x})$ gives an $|\mathbf{x}|$ -by- $|\mathbf{x}|$ diagonal matrix with the elements $\mathbf{x}$ on the diagonal. The quadratic weights $\mathbf{c}_x^q$ are usually equal over the entire optimization horizon (and some algorithms rely on this property). To incorporate $\mathbf{d}_x$ in (3.66), we can choose $\mathbf{c}_x^\ell = -2 \text{diag}(\mathbf{c}_x^q) \mathbf{p}_x^{\text{goal}}$.

In contrast to problem (3.62), problem (3.64) and (3.66) can be straightforwardly composed, which leads to a cost function of the same type (i.e. the sum of two quadratic functions is again a quadratic function with the same structure). By keeping the structure uniform and simple, we can avoid an increase in complexity of the lower nodes in the tree. For example, when a child node has a local goal profile for itself, then we may add the quadratic and linear coefficients of the parent (e.g. neighbourhood) and the child (e.g. house) cost functions to each other (i.e., $(a_1x^2 + b_1x + c_1) + (a_2x^2 + b_2x + c_2) = (a_1 + a_2)x^2 + (b_1 + b_2)x + (c_1 + c_2)$). Conversely, we can translate this composite cost function back to a composite goal profile that represents the goal of both the parent and the child, which is useful for device scheduling algorithms that work directly on quadratic difference profiles, such as e.g. [190].

3.5.3. ALGORITHM

To address the problem(s) specified in Section 3.5.2, we use a greedy heuristic that iteratively asks every child $i \in \mathcal{I}_\mathcal{X}$ to improve the objective of the given parent node $\mathcal{X}$. We already outlined the algorithm and presented an example of its execution in Section 3.5.1, except that this example did not yet account for $z_\mathcal{X}^{\text{down}}$. This heuristic may in principle be used more widely than just for profile steering. However, profile steering allows for a compact representation of the imposed pricing scheme. A more direct description of this algorithm, where the objective considers only the 2-norm distance term, can be found in [HT:6].

First, we generalize the optimization problem of a node $\mathcal{X}$ back to:

$$\min z_\mathcal{X} \quad (3.67)$$

$$z_\mathcal{X} = z_\mathcal{X}^{\text{up}}(\mathbf{x}_\mathcal{X}) + z_\mathcal{X}^{\text{int}}(\mathbf{x}_\mathcal{X}) + z_\mathcal{X}^{\text{down}}. \quad (3.68)$$

The profile steering cost term (from (3.62) in the case of a 2-norm term, or (3.66) in the case of a quadratic distance term, both without $z_\mathcal{X}^{\text{down}}$) is represented as a part of either $z_\mathcal{X}^{\text{up}}$ or $z_\mathcal{X}^{\text{int}}$. Again, we index the variables in (3.68) by superscript (k) , for iterations $k \in \{0, \dots, n_k\}$.

To start the search ($k = 0$), we have to specify some initial solution for all $i \in \mathcal{I}_\mathcal{X}$. We may start with $z_i^{\text{up},(0)} = 0$, but experiments have shown that for profile steering, using a goal profile where every child locally already tries to follow (a fraction of) the profile (i.e. $\mathbf{p}_i^{\text{goal},(0)} = \frac{1}{|\mathcal{I}_\mathcal{X}|} \mathbf{p}_\mathcal{X}^{\text{goal}}$) gives better results (less iterations needed and more evenly spread demand). In practice, the problem may already be solved in many cases before the iteration process even starts. Comparable first-iteration heuristics may also be developed for other cost functions.

In iteration $k \in \{1, \dots, n_k\}$, node $\mathcal{X}$ now asks each of its children i to give an improving pattern for (3.68), i.e. i is asked to solve the complete problem of its parent $\mathcal{X}$, where i can only influence its own variables. To represent this problem for i , we take the solution of the previous iteration $k - 1$, and remove the contribution of i to this solution. We use a subscript sel to refer to the currently selected pattern of i (as opposed to the one proposed by i in iteration $k - 1$). Subsequently, the local

problem of i generates a new solution with a demand $\mathbf{x}_i^{(k)}$ and a subtree cost $z_i^{\text{sub},(k)}$ within this problem. The subtree cost of i in the current solution is represented within $z_{\mathcal{X}}^{\text{down},(k-1)}$, and is reintroduced in the local optimization (and not in its upward cost). This gives the following upward cost function for a child $i \in \mathcal{I}_{\mathcal{X}}$ of $\mathcal{X}$:

$$\begin{aligned} z_i^{\text{up},(k)}(\mathbf{x}_i^{(k)}) &= z_{\mathcal{X}}^{\text{up}}(\mathbf{x}_{\mathcal{X}}^{(k-1)} - \mathbf{x}_{i,\text{sel}}^{(k-1)} + \mathbf{x}_i^{(k)}) + \\ &\quad z_{\mathcal{X}}^{\text{int}}(\mathbf{x}_{\mathcal{X}}^{(k-1)} - \mathbf{x}_{i,\text{sel}}^{(k-1)} + \mathbf{x}_i^{(k)}) + \\ &\quad z_{\mathcal{X}}^{\text{down},(k-1)} - z_{i,\text{sel}}^{\text{sub},(k-1)}. \end{aligned} \quad (3.69)$$

The subtree cost $z_i^{\text{sub},(k)}$ of a candidate solution is added in the local cost function $z_i^{(k)} = z_i^{\text{up},(k)} + z_i^{\text{sub},(k)}$ of a child node i . Together with the definition in (3.69), this leads to a variant of the parent's problem (3.68) where we can change only the assignment of the variables of i . Depending on the internal structure of $z_{\mathcal{X}}^{\text{up}}$ and $z_{\mathcal{X}}^{\text{int}}$, many terms may be constant in relation to $\mathbf{x}_i^{(k)}$. The terms $z_{\mathcal{X}}^{\text{down},(k-1)}$ and $z_{i,\text{sel}}^{\text{sub},(k-1)}$ are always constant, as these describe the solution in the previous iteration. The change to the demand of a child $\mathbf{x}_i^{(k)}$ may also lead to a change in the internal demand $\mathbf{x}_{\mathcal{X}}^{\text{int},(k)}$ of $\mathcal{X}$, e.g. due to grid losses. However, this change is generally negligible and may be ignored (it is accounted for in subsequent iterations). Every child $i \in \mathcal{I}_{\mathcal{X}}$ now optimizes its own local cost $z_i^{(k)}$ using the upward cost (3.69), and determines a response pattern $\langle \mathbf{x}_i^{(k)}, z_i^{\text{sub},(k)} \rangle$. Using the outcome of the optimization, we determine the improvement $\tilde{z}_i^{(k)}$ that each child can make to the objective of $\mathcal{X}$:

$$\tilde{z}_i^{(k)} = z_i^{(k)} - z_{\mathcal{X}}^{(k-1)}. \quad (3.70)$$

We select the “winning” child $i^{(k)}$ that makes the largest (most negative) improvement, which is defined by:

$$i^{(k)} = \arg \min_{i \in \mathcal{I}_{\mathcal{X}}} \tilde{z}_i^{(k)} \quad \forall k \in \{1, \dots, n_k\}. \quad (3.71)$$

The search is terminated if no improvement is found or the improvement is only marginal. We also terminate the search after a maximum of n_k iterations have been carried out.

The local cost $z_i^{(k)}$ describes both the cost of i and the effect it has on the cost of its parent $\mathcal{X}$ (through (3.69)). Note that in principle we may choose to let the winning child $i^{(k)}$ not participate in iteration $k+1$, as it should not be able to give a further improvement if the problem is solved to optimality. We subsequently update the pattern selection of the children of $\mathcal{X}$, i.e. we set only the pattern of child $i^{(k)}$ to the new pattern it proposed. Based on this, we can determine $\mathbf{x}_{\mathcal{X}}^{(k)}$ and $z_{\mathcal{X}}^{\text{down},(k)}$ from the combination of the selected patterns for all children.

If the problem of $\mathcal{X}$ has a simple sum structure (such as for profile steering), then we can update the selection efficiently as follows:

$$\mathbf{x}_{\mathcal{X}}^{(k)} = \mathbf{x}_{\mathcal{X}}^{(k-1)} - \mathbf{x}_{i^{(k)},\text{sel}}^{(k-1)} + \mathbf{x}_{i^{(k)},\text{sel}}^{(k)} \quad \forall k \in \{1, \dots, n_k\} \quad (3.72)$$

$$z_{\mathcal{X}}^{\text{down},(k)} = z_{\mathcal{X}}^{\text{down},(k-1)} - z_{i^{(k)},\text{sel}}^{\text{down},(k-1)} + z_{i^{(k)},\text{sel}}^{\text{down},(k)} \quad \forall k \in \{1, \dots, n_k\} \quad (3.73)$$

This concludes the description of the algorithm.

In the case of profile steering, the cost function that we present to the children can be represented with a profile $\mathbf{p}_i^{\text{goal}}$ and a weight $w_{p^{\text{goal}},i} = w_{p^{\text{goal}},\mathcal{X}}$, as discussed in Section 3.5.2. The expression (3.69) is based on using the following profile (excluding the terms $z_{\mathcal{X}}^{\text{down},(k-1)}$ and $z_{i,\text{sel}}^{\text{sub},(k-1)}$):

$$\mathbf{p}_i^{\text{goal},(k)} = \mathbf{p}_{\mathcal{X}}^{\text{goal}} - \mathbf{x}_{\mathcal{X}}^{(k-1)} + \mathbf{x}_{i,\text{sel}}^{(k-1)}. \quad (3.74)$$

3.5.3.1. DECENTRALIZED VERSION

To make an efficient decentralized implementation of this algorithm, we can exploit some properties of the algorithm in general, and of its application for profile steering specifically. In every iteration, we have to send the objective and constraints of $\mathcal{X}$ to all children, with some slight changes for every i and for the current pattern selection (usually just $\mathbf{x}_{\mathcal{X}}$). If each child i can perform these changes locally, then we may broadcast the problem description, which may be implemented much more efficiently than sending individual messages to all children. In the case of profile steering, we only have to broadcast $\mathbf{d}_{\mathcal{X}}^{(k-1)} = \mathbf{p}_{\mathcal{X}}^{\text{goal}} - \mathbf{x}_{\mathcal{X}}^{(k-1)}$ in each iteration k . Each child knows its own values of $\mathbf{x}_{i,\text{sel}}^{(k-1)}$ and $z_{i,\text{sel}}^{\text{sub},(k-1)}$, and can adapt the local problem accordingly. Furthermore, we only need to consider the changes in the problem between consecutive iterations: for profile steering, this means that we may broadcast $\mathbf{d}_{\mathcal{X}}^{(k-2)} - \mathbf{d}_{\mathcal{X}}^{(k-1)}$ for $k > 1$ (however, this makes the approach less robust against message loss, so we recommend to use this approach only for large structures in the first iteration). After this, to determine the winner $i^{(k)}$, each child can locally determine $\tilde{z}_i^{(k)}$, and send only this cost improvement to $\mathcal{X}$. We may use a tree structure to collect the largest cost improvement in $\mathcal{X}$, and may furthermore omit the response if a child does not give any (significant) improvement. Next, we notify $i^{(k)}$ that it has provided the winning solution (which now becomes the selected solution), collect the pattern of the winning solution (or the difference to its previous selected solution), and update the problem. In principle, in case of a simple sum structure, the winner may broadcast $\mathbf{x}_i^{(k)} - \mathbf{x}_{i,\text{sel}}^{(k-1)}$ directly to its siblings to adapt their upward cost function, i.e. $\mathbf{p}_i^{\text{goal},(k)} = \mathbf{p}_i^{\text{goal},(k-1)} + \mathbf{x}_{i^{(k)},\text{sel}}^{(k-1)} - \mathbf{x}_{i^{(k)}}^{(k)}$ in the case of profile steering.

3.5.3.2. PASS-THROUGH NODES

For profile steering, we may define a variant of the pass-through node that we defined in Section 3.2.5.3 that may be used to split off (and separately optimize) subgroups of children. In principle, this is just a profile steering node that does

not add any internal costs, and simply optimizes the profile steering problem of a parent using the children that have been assigned to it. The node in principle has limited special properties in comparison to Section 3.2.5.3, yet does allow for the local (pre-)optimization of a subproblem. In a nested context, we generally reuse the solution that was generated in the previous iteration of the parent problem as a starting point for the search, similar to the pattern reuse in a nested context for column generation (see Section 3.4.4.5).

3.5.3.3. USING MULTIPLE PROPOSED IMPROVEMENTS

One of the drawbacks of the presented approach is that we should in principle only accept the improvement of a single child in every iteration, which means that the number of iterations may become proportional to the number of children. To avoid this, we propose to apply the improvements of multiple children in a single iteration. Applying multiple patterns means that it is no longer guaranteed that a child pattern leads to an improvement: the patterns may contribute to the same part of the demand profile, which may lead to overshoot. We avoid this by evaluating the improvement ourselves.

The modified procedure works as follows. We order the proposed patterns by improvement (relative to the previous assignment), and incrementally apply these patterns until the application of a pattern would lead to a worse aggregate profile. The experiments that we perform in Section 3.5.4 show that this modification can significantly reduce the number of iterations in the search process. However, in a distributed (and non-nested) context it may still be more expensive in terms of communication than the original method, as we have to transport more profiles from the children. We see in the experiments that, especially in the first iterations, the application of multiple patterns leads to new peaks that have to be corrected afterwards. A more balanced approach may be to apply only a few (e.g. at most 10) patterns in each iteration. Also, we may try to make the iterations more effective by dividing up the difference profile among the children (see Section 3.5.5.1.1).

3.5.4. EXPERIMENTS

We perform some small experiments on a large case to illustrate the behaviour of profile steering, using a set-up comparable to the one we used to demonstrate the IDDP method (Section 3.3.5). Later in Section 3.6, we consider a larger simulation study where we compare the presented methods. More experimental results for profile steering may be found e.g. in [HAT:6].

The set-up that we consider in this section is illustrated in Figure 3.20. At the top level we have a profile steering node that controls 400 households. As the households do not have internal costs, we may use profile steering pass-through nodes at the household level. We consider a flat profile (0 W) as the goal profile of the root node. The weight $w_{p_{\text{goal}},N}$ of the balancing component is not relevant as neither the modelled devices nor the households have internal costs.

FIGURE 3.20: Node configuration of profile steering in experiments.

FIGURE 3.21: Planned demand during iterations of profile steering planning session at 12:00 on the first day of Flex Street.

We consider two variants of the profile steering algorithm: the basic variant that selects only one of the proposed pattern changes per iteration, and a variant that applies the proposed changes in the order of expected improvement until this no longer leads to an improvement (see Section 3.5.3.3). In both cases, we use the quadratic formulation of the problem from Section 3.5.2.3. We run the profile steering algorithms until we no longer find an improvement at both the root level and the household level ($k_x = k_i = \infty$ for $i \in \mathcal{I}_x$).

We present the results of the simulation for the planning session at 12:00 on the first day (as described in Section 3.3.5) in Figure 3.21. In this figure, the results are given for both the case where we apply only a single improving pattern (Figure 3.21a) and the case where we apply the proposed changes until this no longer leads to an improvement (Figure 3.21b). In both cases, comparing the results to Figure 3.8 (p.73, $k_x = 0$) we see that the demand peak in the initial iteration of the root node is already significantly lower than for IDDP and column generation. The reason for this is that the house level profile steering problem locally already tries to produce a flat demand profile. The presented results show that they were quite successful.

In the first case (apply only the best update, Figure 3.21a), the progress after the first iteration is slow, because we can update the pattern of only a single household subproblem in each iteration, which means that we may need around 400 iterations (one for each subproblem) to optimize the objective. Although the first 50 iterations give a reasonable decrease in peak demand by using the households with a lot of flexibility, the progress after this becomes very slow as we also can shift the demand of the households with less flexibility. However, it is unwise to simply ignore these households, as the iterations after $k_x = 50$ still represent 10% of the available peak shaving potential. After 380 iterations, we find a very smooth aggregate demand profile. Note that even though this aggregate profile is smooth, this does not necessarily mean that the individual demand profiles are smooth as well (if this household level smoothness is relevant for the case at hand, then we should add internal costs to account for this). The number of iterations is high and may be considered to be impractical, which has lead to the idea to apply multiple proposed improvements in the same iteration.

In this second case (apply updates until worse solution, Figure 3.21b), the convergence is much faster, especially in the first iterations. As for the column generation results presented in Section 3.4.4.2, the distance to the final profile roughly halves in each of the first iterations, although it does introduce artifacts in the intermediate aggregate demand profile. The reason for this is that the application of multiple changes leads to an overcompensation for the highest difference values. The idea to apply multiple proposed changes becomes less effective in later iterations: once the largest improvements have been made, the overlap between the improvements becomes high in some less constrained time intervals, which means that less profiles may be applied together. The search for the given case is completed after 65 root node iterations, finding almost the same aggregate profile as the first variant.

3.5.5. EVALUATION

Profile steering is a simple and highly effective optimization method that is easy to implement. The underlying local optimization algorithm of the method is very flexible. Individual iterations of the method are very efficient from a communication and computation perspective (we have also implemented this approach on a lightweight embedded platform). However, the current approach still needs a lot of iterations to find a good solution, even though we have developed an extension to significantly reduce the number of needed iterations. Therefore, the recommendations focus on efficiency improvements that reduce the iteration count.

Looking at the results of the optimization, we may consider the results to be “unfair” to devices and households with little flexibility, as the approach prioritizes nodes that have the most flexibility available. If this is considered a problem, then this should be expressed in the objective.

3.5.5.1. RECOMMENDATIONS

- 3.5.5.1.1. *Split the difference* The current approach asks every node to realize the complete difference profile, which is completely unrealistic in the first iterations of the problem. A more realistic approach would be to split the goal profile between nodes such that every node realizes contributes to a small part of the difference profile, as was e.g. also proposed by Bakker [19: p. 71]. As the search proceeds, we may ask the nodes to contribute a larger part of the difference profile, until nobody is able to contribute an improving profile.
- 3.5.5.1.2. *Self-reflection* We may try to anticipate on the profiles that others may generate, such that the problems generate a diverse set patterns that can be combined. To this end, we may first locally apply the proposed difference profile, and then schedule the device on top of this difference profile. This procedure may be repeated several times to estimate the response of (some) other devices in the following iterations. A device should return either all of these improvements or only a randomly chosen one of these improvements to its parent, with the idea that the patterns of the children can together lead to useful combinations that already anticipate the results that would be found in following iterations in the original approach.
- 3.5.5.1.3. *Worst fit allocation* As we spend most of the iterations to resolve the coordination of the smaller problems, in the first iterations an approach similar to the *worst fit* memory allocation technique [297: p. 287] may be useful to reduce the fragmentation of the flexibility. In this case, we propose to consider the sequence of possible improvements in reverse order (while considering only the profiles that indeed give an improvement), such that we first consider the problems that have less flexibility available. We should be able to apply many more of these small changes before overshoot occurs. This allocation strategy should first put the smaller loads approximately in the right part of the demand profile, and then leaves the larger flexibility available to resolve the final gaps. This implies that we should switch back to the original strategy after a few iterations to exploit the larger subproblems.

3.6. YEAR SIMULATION EXPERIMENTS

3.6.1. INTRODUCTION

In this chapter, we have developed planning algorithms for (subsets of) the model that was developed at the beginning of this chapter. In this section, we compare these planning algorithms in a common simulation scenario. We use the Flex Street case (Moderate variant), which is described in Appendix B and has been developed in a comparative evaluation study with TRIANA in [HAT:8]. The case describes a

diverse set of 400 households with several flexible devices based on realistic data for the duration of a year. The objective for the case is to flatten the demand profile. With the duration of a year, we can expect the DSM method to come across a wide range of different situations, and thereby we are able to validate that an approach functions not just for a single environment within it has been developed.

As an objective reference, we have determined a model for the Flex Street case in Appendix B.6, which makes an ADP-like overapproximation of the available flexibility. With this model, we can set up a *quadratic programming* (QP) model for the aggregate demand of all devices, which gives a lower bound on the sum of square demand values, and thereby on the RMS value. The RMS value is used to indicate the losses in the grid. We have shown in Appendix B.6.3.4 that the peak demand of the aggregate solution is a lower bound on the peak demand. Note that this model only describes the behaviour at an aggregate level, and can not (efficiently) describe household level phenomena. In addition, to evaluate the ramp rate of the demand, we have set up a aggregate model that minimizes the sum of squared differences between consecutive demand values (Appendix B.6.3.5).

In contrast to the earlier mentioned study in [MΓ:8], forecast errors are not accounted for in the present experiments, as e.g. a high frequency replanning is expensive to be used for a year simulation. We acknowledge that these errors can have a significant effect on the results. We discuss possible alternatives in Section 3.7.1, and present a planning method that is more robust to forecast errors in Chapter 4.

We present the experiments as follows. First, we give the configuration of the scenario, the simulation, and the optimization methods in Section 3.6.2. We present and describe the simulation results in Section 3.6.3, followed by an evaluation in Section 3.6.4. After this, we conclude this chapter in Section 3.7.

3.6.2. CONFIGURATION

3.6.2.1. SCENARIO

- We use the Flex Street scenario (Appendix B), and from this scenario we use the Moderate variant.
- For the lower bound model from Appendix B.6, we use the variant with the quadratic balancing objective (Appendix B.6.3.4).
- Both the used lower bound model from Appendix B.6 and used the device level DP models do not account for thermal buffer losses over time.
- We choose not to use the provided prediction information, and instead use a clairvoyant prediction.
- The simulation does account for battery charging losses.
- The devices in Flex Street do not have internal costs.
- The models describe the cost in terms of the demand at an aggregate level, and do not describe the cost for the electricity demand at a household level.

3.6.2.2. SIMULATION ENVIRONMENT

All experiments in this section are performed on a desktop PC with an Intel Core i5-6600 (4 cores, 16 GB of RAM). Since the scope of the simulations is quite extensive, we use the C++ simulation environment from [231: Chapter 5] to perform the experiments, which has been modified to support a dynamic hierarchical modelling according to Section 3.2 and to suppress the behaviour of their real-time control mechanism. The experiments use more accurate and efficient DP formulations than those used for the experiments in [$\mathbb{H}\mathbb{T}$:3, $\mathbb{H}\mathbb{T}$:4, $\mathbb{H}\mathbb{T}$:8]. All experiments use the same device level DP optimization models, and a control strategy that dispatches devices according to what is specified in the planning. The column generation method and the aggregate lower bound QP use CPLEX 12.5.

3.6.2.3. OBJECTIVE/KEY PERFORMANCE INDICATORS

Flex Street in principle prescribes a cost function with quadratic terms for the household level demand and the aggregate demand of the group of houses, and a complicated term with multiple exponential terms that depends on the “peak demand”⁸ of the group [$\mathbb{H}\mathbb{T}$:8: p. 265]. As this cost function is difficult to deal with, we use a different set of evaluation criteria (where $\mathbf{x}_x$ gives the aggregate demand of the neighbourhood for the whole year, i.e. 35040 time intervals):

- Aggregate demand values $\mathbf{x}_x$

- Maximum over the year: $\max \mathbf{x}_x$. (3.75)

The maximum demand value is a representative for the investment costs that have to be made to support the aggregate demand of the neighbourhood.

- Minimum over the year: $\min \mathbf{x}_x$. (3.76)

The minimum demand value may in principle be a representative for the degree of self-consumption of PV (higher is better), although the moments where the PV feed-in exceeds the uncontrollable demand are quite limited in the case at hand.

- Average over the year: $\frac{1}{35040} \sum \mathbf{x}_x$. (3.77)

The average value is a representative for the device losses that are incurred by the control due to the use of the battery and the auxiliary resistance heating rod. These losses lead to extra electricity commodity costs, and also contribute to the other criteria.

- RMS value: $\sqrt{\frac{1}{35040} \mathbf{x}_x^\top \mathbf{x}_x}$. (3.78)

The RMS value gives the effective quadratic value of the demand, and is thereby a representative for the grid losses at an aggregate level.

8. The article defines “peak demand” as the demand value that covers the demand over 99.73 % of the time, i.e. 364 out of 365 days. We simply use the maximum demand value, i.e. $\max \mathbf{x}_x$ based on 15 min averages.

- Ramp rate $x_{\mathcal{X}}(t) - x_{\mathcal{X}}(t-1)$ per 15 min

- Maximum over the year: $\max_{t \in \{2, \dots, 35040\}} (x_{\mathcal{X}}(t) - x_{\mathcal{X}}(t-1)). \quad (3.79)$

- Minimum over the year: $\min_{t \in \{2, \dots, 35040\}} (x_{\mathcal{X}}(t) - x_{\mathcal{X}}(t-1)). \quad (3.80)$

The most positive and negative ramp rate are relevant for conventional generation resources in particular, as these have a limited ramp rate: large thermal plants prefer to (or are only able to) change their output level only slowly (closer to zero is better).

- Average absolute value: $\frac{1}{35040} \sum_{t \in \{2, \dots, 35040\}} |x_{\mathcal{X}}(t) - x_{\mathcal{X}}(t-1)|. \quad (3.81)$

Although large demand changes may lead to constraint violations, smaller demand changes should in principle be avoided as well for efficiency reasons, although this may become less relevant when the role of DSM based storage increases.

The ramp rate always has to be considered over a given time period, as it gives the rate of change over time of the demand value (in principle the derivative of $x_{\mathcal{X}}(t)$). In the experiments, we use values over 15 min periods as we have no further information.

We choose not to provide weights for these different criteria (although we have been forced to specify weights within the column generation approach). Note that none of the presented optimization methods is able to jointly optimize all of these terms.

There are no constraints on the electricity demand values at neither the household level or the aggregate level. The device models have internal constraints.

3.6.2.4. OPTIMIZATION METHODS

In the following, we discuss the parameters for the specific planning methods that we used in the simulation experiments. In all cases, we use a time interval length of $\tau = 15$ min, a planning interval of 6 h (24 time intervals) and an optimization horizon of 24 h (96 time intervals). We have not implemented the reuse of planning results from previous planning sessions in the used simulation environment.

- 3.6.2.4.1. *IDDP* We use the node configuration of Figure 3.7 (p. 72), which has a single IDDP group node at the top level and pass-through nodes at the household level. To give the procedure sufficient time to converge, we choose $n_k = 30$ (although little progress is made after $n_k = 15$). We furthermore choose not to change the default parameters for IDDP, i.e. we start at a price level of $\mathbf{c}_{\mathcal{X}}^{\ell} = \langle 1000, \dots, 1000 \rangle$, and use a price change step size of $-c_i^{\text{down},(k)} = c_i^{\text{up},(k)} = 75$, and the goal profile is derived from the energy content of the first iteration in each planning session.

- 3.6.2.4.2. *Column generation* We use the node configuration of Figure 3.10 (p. 96), using a single column generation master problem at the top level and pass-through nodes at the household level. The master problem uses the modelling of the objective that we introduced in Section 3.4.4.1 and the rounding-based column selection method, with two rounding phases ($n_{k,\mathcal{X},\mathcal{I}} = \langle 10, 2 \rangle$).
- 3.6.2.4.3. *Profile steering* We use the node configuration of Figure 3.20 (p. 115), using a profile steering master problem as the top level, and subordinate profile steering problems at the household level. We use the profile steering variant that can apply multiple patterns in a single round. The household profile steering problems are configured not to contribute internal cost to the problem. We do not limit the number of iterations ($n_{k,\mathcal{X}} = \infty$ and $n_{k,i} = \infty$ for $i \in \mathcal{I}_{\mathcal{X}}$). In the experiments, the optimization generally terminates in 50 – 60 iterations, and the local optimization terminates in 2 – 4 iterations. Note that the last 30 – 50% of top level iterations hardly improve the demand profile, and may therefore be omitted.

3.6.3. RESULTS

3.6.3.1. LOAD DURATION CURVES

To evaluate the results of the simulations following the setup of Section 3.6.2, we present the load duration curves of the simulations with the different optimization methods over the year in Figure 3.22, together with the load duration curve for the aggregate lower bound. An ideal load duration curve would be a horizontal flat line. At a first glance, the figure appears to tell that each of the control methods gives roughly the same distribution of load over the year, which is to some extent true. The main reason for this is that the devices offer only short-term flexibility (e.g. in a range of 4 – 24 h), and can thereby not shift demand between seasons. Next to the realized loads, also the aggregate lower bound (which is representative for the best attainable load duration curve) is included in the figure. Based on this curve, we may conclude that the results seem to be near to the optimum. Most of the improvement is made when we introduce at least *some* form of coordination, where we consider the situation without coordination as a baseline (see Appendix B.4, Figure B.3 on p. 289).

- 3.6.3.1.1. *Highest demand values* However, if we look more closely, we see that there are still notable differences between the methods. Figure 3.22b focuses on the highest demand values of the year, which lead to the most strain on the electricity system. These values are reached in winter, where there is high use of the heat pumps and low PV feed-in. Here we see that profile steering comes very close to the lower bound solution (a more quantitative evaluation follows). The load duration curves of both the profile steering solution and the lower bound solution are not smooth due to plateaus with the same demand value in their demand curves (see Figure 3.8c or later Figure 3.24). The distance that remains between these solutions may be caused by an overapproximation of the available flexibility by the lower bound model. The column generation method follows closely behind it, but performs less

well when it comes to a more general reduction of high load values (to the right of the highest peak value). IDDP leads to peak demand values that are notably (20 kW) higher than those for profile steering and column generation. Looking back at Section 3.3.5 (or forward to Figure 3.24), a lot of this difference may be explained by the substantial noise that results from the randomized price update rule. Although this improvement is not as large as we may have hoped for, it is still economically relevant. The (less large) noise also explains most of the difference in peak value between the column generation approach and the profile steering approach if we consider Figure 3.8b (or Figure 3.24).

3.6.3.1.2. Lowest demand values The differences are larger if we consider the lowest demand values over the year, which we focus on in Figure 3.22c. These values are reached in summer, when there is low use of the heat pumps and high PV feed-in. In this plot, higher values are better, as long as this contributes to a decrease in demand at other times with a higher demand. We see that the aggregate lower bound gives notably better values here than the other methods. A possible explanation for this difference is that this approach can look further ahead than the realized methods, which use an optimization horizon of a day, and some devices have a flexibility range of more than a day. For example, the heat pumps may have a flexible range of 36 h in summer as there is only limited heat demand (Appendix B.6.3.1.5), and a suitable distribution over the days may significantly increase the lowest demand value. The optimization methods have an incentive to minimize the energy use only within the considered optimization horizon, and may therefore choose to shift demand beyond the horizon even if substantial PV feed-in is available. The aggregate model may also exploit some flexibility that is not available in practice, e.g. from splitting up a washing machine profile. The column generation method performs worse in this respect, which may be explained by conflicting objectives (in an earlier version where we did not yet account for the costs to buy electricity [HT:3], the lowest value was “improved” by introducing substantial losses).

3.6.3.2. RAMP RATE DURATION CURVES

In Figure 3.23, we consider the load duration curves of the ramp rates of the demand in the simulations. In this plot, we also include the results for the aggregate model where we optimize the sum of square differences (marked diff. agg. bound; see Appendix B.6.3.5). Note that only this model and the column generation formulation explicitly incorporate the ramp rate aspect in their objective. However, it seems that the column generation approach is fairly ineffective in reducing the ramp rate, because other competing (and conflicting) objectives are considered to be more important. Also, the noise on (i.e. continuous fluctuation of) the profile leads to many small demand changes over time, similar to (but less extreme than) IDDP. Both the aggregate lower bound solution and the profile steering solution give sudden large changes in the demand value, which may give operational challenges if we would apply this solution directly at a larger scale in practice.

(a) Complete year.

(b) Highest demand.

(c) Lowest demand.

FIGURE 3.22: Load duration curves with different planning methods.

FIGURE 3.23: Difference duration curves with different planning methods.

FIGURE 3.24: Demand profile around critical peak period.

approach	demand				ramp rate			sim./opt. time
	max.	min.	avg.	RMS	max.	min.	avg.	
IDDP	490.3 kW	80.2 kW	255.6 kW	264.9 kW	93.3 $\frac{\text{kW}}{15\text{min}}$	-79.9 $\frac{\text{kW}}{15\text{min}}$	8.7 $\frac{\text{kW}}{15\text{min}}$	2 h 47 min
col. gen.	470.1 kW	60.1 kW	256.0 kW	265.0 kW	74.9 $\frac{\text{kW}}{15\text{min}}$	-32.3 $\frac{\text{kW}}{15\text{min}}$	5.5 $\frac{\text{kW}}{15\text{min}}$	4 h 51 min
profi. st.	467.1 kW	81.3 kW	254.5 kW	263.1 kW	115.2 $\frac{\text{kW}}{15\text{min}}$	-114.1 $\frac{\text{kW}}{15\text{min}}$	1.7 $\frac{\text{kW}}{15\text{min}}$	14 h 50 min
agg. bnd.	463.0 kW	134.9 kW	254.4 kW	262.0 kW	122.0 $\frac{\text{kW}}{15\text{min}}$	-115.0 $\frac{\text{kW}}{15\text{min}}$	0.7 $\frac{\text{kW}}{15\text{min}}$	3 s
agg. diff.	473.4 kW	128.8 kW	256.7 kW	264.7 kW	10.8 $\frac{\text{kW}}{15\text{min}}$	-7.5 $\frac{\text{kW}}{15\text{min}}$	0.7 $\frac{\text{kW}}{15\text{min}}$	8 s

TABLE 3.1: Year simulation results with Flex Street (Moderate variant).

3.6.3.3. CRITICAL WEEK PROFILES

We present the demand profiles for the week with the highest demand values, which occurs near the end of the year in winter, in Figure 3.24. This shows that there are still substantial differences between the demand profiles that are produced by the control methods, even though the peak demand is comparable. At some places, we also see some artifacts resulting from the 6 h planning interval (corresponding with the minor ticks of the horizontal axis). The behaviour at some places changes sharply after a replanning, which may indicate that we can benefit from an optimization horizon longer than 24 h, an estimation of cost behind the horizon, or from more frequent replanning. This applies in particular to IDDP, which determines a new target profile based on the expected average demand in each planning session. The profile steering result tracks the aggregate lower bound solution quite well (with a slight positive offset), although this approach also tries to postpone demand beyond the horizon in some cases. The noise on the profile substantially increases the peak demand value in particular for IDDP, and to a lesser extent for column generation. For the column generation approach, we see that the optimization realizes a smooth decrease in demand after the large evening demand peaks. However, this comes at the expense of consumption capacity to fill the PV valley in the afternoon of each day (note that the PV valley is relatively small in the considered example as we consider days in winter).

3.6.3.4. QUANTITATIVE RESULTS

In Table 3.1, we present the quantitative results of the simulations, i.e. the values of the objective components that we present in Section 3.6.2.3.

- 3.6.3.4.1. *Peak demand* We see the peak (max., left part of table) demand as the leading criterion. On this criterion, the column generation and profile steering come close to the aggregate lower bound, with a 1.5% and 0.9% (profile steering) higher value. IDDP reaches a value that is 5.9% above the bound, although we have used a higher than usual number of iterations to reach this result. Note that the result without coordination, as presented in Figure B.3, gives a 135.4% higher peak demand. Although the peak reduction from the improved optimization methods is far smaller than when we consider the uncoordinated situation as a baseline, a 5 percentage

point improvement can still lead to a significant decrease in costs. An interesting point is that the approach considering the sum of squared differences in the aggregate lower bound model (agg. diff.; Appendix B.6.3.5) also leads to a reasonable peak demand, even though we have not explicitly asked for this in the objective. The reason for this is that the total energy content of the problem is more or less fixed, and the optimal solution for the balancing problem also leads to a flattening of the difference values (but then with jumps between the flat demand values).

- 3.6.3.4.2. *Lowest demand* The lowest observed demand values (min., left part of table) show quite some differences between the approaches. As discussed before in Section 3.6.3.1.2, each of the presented approaches performs significantly worse than the reference aggregate lower bound model due to limitations in the rolling horizon approach, and a possible overapproximation of flexibility in the aggregate lower bound model. The column generation approach fares less well in particular, as it considers the other objectives to be more relevant.
- 3.6.3.4.3. *Average demand* The average of the demand values (avg., left part of table) represents the total electricity demand, and is therefore representative for the losses in the devices (which are caused by battery losses and use of the auxiliary resistance heating rods). The model does not consider grid losses as a part of these losses. We see that IDDP and column generation lead to a notable increase in average demand. The reason for this is that both methods make excessive use of the battery, which is caused in part by overcompensation due to the linear pricing scheme to find the desired demand value. Furthermore, the used objective formulation for the column generation approach results in the use of the battery to make the demand profile more smooth. These losses may reduce the profit from the control approach: each kW of extra control related demand at a neighbourhood level may e.g. cost €438 per year at a neighbourhood level (if we consider a commercial electricity of tariff €0.05/kWh), i.e. €1.10 per household per year. Profile steering finds a solution that is very close to the aggregate lower bound solution.
- 3.6.3.4.4. *RMS value of demand* For the RMS value of the aggregate demand we see a similar trend as for the average demand: an increase of the average demand value naturally leads to an increase of the RMS value. The noise on the column generation profile is lower than the noise on the IDDP profile, and therefore the RMS value increases less than the difference in average value between IDDP and the column generation result would suggest. The profile steering method performs marginally worse on this metric (0.4%) than the aggregate lower bound model, as it may in some cases have less freedom to distribute the demand (see Section 3.6.3.3).
- 3.6.3.4.5. *Highest ramp rate* The highest ramp rate (max., right part of table) determines the highest (upward) change in consecutive demand values, and may be relevant in particular when the system is integrated in a context with conventional power plants (lower is better). We see that each of the presented methods, except for the approach that is specifically optimized to avoid this (agg. diff.), lead to high upward ramp rates. This also applies to the column generation method, which incorporated the maximum difference value as part of its objective. However, a

decrease in ramp rate comes at the expense of an increase in peak demand, and the latter is considered to be more expensive. Also, the profile noise may in some cases contribute to the difference peak.

- 3.6.3.4.6. *Lowest ramp rate* The lowest (downward) ramp rate (min., right part of table) has a similar relevance as the highest ramp rate (closer to zero is better). We find similar high values as for the highest upward ramp rate. For the column generation method, the reduction of the highest downward ramp rate does not appear to conflict with the peak minimization objective, but the shape of the demand profile that we have seen in Section 3.6.3.3 and the results of Section 3.6.3.4.2 do suggest that the more smooth profile comes at the expense of a decrease in the lowest demand level. The profile noise may affect the minimum value as well.
- 3.6.3.4.7. *Sum of absolute differences of ramp rate* The sum of absolute differences of the ramp rate ($|\text{avg.}|$) indicates the variability of the demand. A higher variability may lead to extra losses in a context with conventional load following power plants. Also, a varying demand value is often suboptimal if we consider a quadratic objective (e.g. the RMS value, which is representative for transport losses). The profile noise of the IDDP and column generation approaches leads to high values for this metric. The value for the profile steering approach is higher than for the aggregate lower bound model as the replanning sometimes leads to a revision of the demand value, but is still much lower than for the IDDP and column generation approaches.
- 3.6.3.4.8. *Simulation/optimization time* The simulation/optimization time (sim./opt. time) describes the amount of computation effort that the optimization problems cost. Despite that we have used a quite high number of iterations for IDDP ($n_k = 30$), it is still by far the fastest of the considered approaches. The column generation approach spends a significant amount of work in the master problem. The profile steering approach is significantly slower than column generation and IDDP, because it has to spend more iterations to optimize the problem, and we also have to solve a subsequent nested profile steering problem at the household level. Still, even though the computation times are *longer* for the more sophisticated methods, this does not mean that the computation times are actually *long*, considering that we simulate a complete neighbourhood for the duration of a year on a single computer. The highly simplified aggregate models of a complete year can be optimized in seconds (but may not correspond with an actual existing solution).

3.6.4. EVALUATION

Based on the simulation results, we note the following points:

- The year-long simulation aims to exclude the element of targeted optimization against a specific case, or just luck. If we look in more detail, we see that there are some critical times in the year, where slight configuration changes may change the outcome of the simulation. For example, the peak demand value for the column generation method may change by ± 2 kW (0.5% of the total demand) if we execute the simulations with slightly different parameters. Also, the basic structure of the weeks in the dataset is quite similar, even

though the parameters (e.g. the actual amount of heat demand) vastly differ. An extensive dataset, including real-world data on the flexibility that can be provided by flexible loads, may help to give a more realistic and detailed view of the performance of DSM methods.

- We have seen in the results that good profiles are not necessarily smooth. The reason for this is that more smooth demand value changes may lead to a waste of energy flexibility. However, for operational reasons it may still be desirable to aim for a smooth profile.
- IDDP aims for a flat demand profile over the complete optimization horizon. However, this goal is far from being reachable with the given flexible resources. We might be able to improve its planning results if we are willing to revise the bounds once it becomes clear that the existing bounds are not attainable.
- Even though column generation and profile steering lead to substantially increased computation times, the planning results are still available in reasonable time. The planning takes 7 – 37 s per top-level planning session on average for the case of Section 3.6 with a centralized scheduling of 400 households. In a decentralized implementation, the computational power scales with the households, and thereby allows the household (and device) problems to be solved in parallel. This helps for the IDDP and profile steering problems, where the time for solving the subproblems is dominant. In contrast to this, for the presented column generation approach, the master problem gives a substantial amount of work, and therefore the distribution of work may be less effective due to Amdahl's law (we may shift work towards the subproblems by admitting less columns in each iteration).

3.6.4.1. RECOMMENDATIONS

Based on the results of the Flex Street year simulation case, we give the following recommendations for a more extensive evaluation.

- 3.6.4.1.1. *Focus on critical times* Year simulations are quite expensive in terms of simulation time, whereas most years only have a few interesting moments that determine the key outcomes, and DSM resources only have a short memory (usually less than one day). Therefore, it may be more effective to focus on these important moments, together with a random sample of short periods scattered over the year to validate that the results are still good at normal times.
- 3.6.4.1.2. *Account for uncertainty* The presented simulations assume a deterministic problem and do not account for uncertainty, e.g. from the amount and timing of PV feed-in, the uncontrollable loads, the availability and constraints of the controllable loads, the heat demand, and the realized efficiency of devices. Although the simulations that we performed earlier with TRIANA did consider these sources of uncertainty (see Appendix B.4), the considered model is fairly limited and the used

control strategy has a limited effectiveness. Furthermore, the presented optimization methods are in their current form less suitable for uncertain environments, and would ask for a lot of reoptimization sessions, which might let the year simulation take about 24 times longer than it already does. Therefore, a more complete evaluation be carried out to evaluate Flex Street with a more representative uncertainty model. For this, a suitable strategy has to be developed to address uncertainty in an efficient and effective way within TRIANA.

- 3.6.4.1.3. *More problems with local costs* In this large simulation case, we have not accounted for local costs of devices and households (although some devices do have losses that lead to extra demand). Flex Street does not model these costs at a device level, and has a limited model for the costs at a household level. Also, the presented aggregate lower bound model for the case does not represent these local costs. A more complete evaluation should include these costs, especially for cases where the subproblems have an objective that conflicts with the objective of the master. Evaluations on a variant of the problem where local costs are relevant are given in Section 3.4.4.5, and results for a different problem where these costs are relevant are given in Section 5.5.3.

3.7. CONCLUSION

In this chapter, we developed more sophisticated optimization algorithms in the line of TRIANA (Section 2.4), leading to a more effective optimization of DSM resources. These methods have more expressive power and give better results than the planning algorithm that was used before. We have set up a common hierarchical model for these optimization algorithms. The developed algorithms have different characteristics: IDDP is a highly scalable heuristic, column generation is a very flexible approach, and profile steering leads to very smooth demand profiles. We compared these algorithms with each other and with a lower bound solution in a large scale simulation case. For the demonstrated (deterministic) Flex Street case, both column generation and profile steering give near optimal results, within 1 – 2% of the lower bound.

These new algorithms lose some efficiency in computation and communication, which are well justified against the efficiency improvements in the energy domain. The results of the planning still can be determined in reasonable time (7 – 37 s per planning session on average for the large case of Section 3.6 with centralized scheduling). In a nested configuration, the approach has some scalability problems, yet the approach may scale well enough if the hierarchical structure is not too deep and the iteration count can be kept low. A local optimization that takes the needs of the higher level problems into account is a key technique to avoid unnecessary communication. We argue that the control step should not be completely separated from the planning step, but should rather be considered as the interface by which the planner can control the behaviour of devices. We develop this idea further in Chapter 4.

One of the choices in the approach is the size of the groups that are joined in a single optimization node. If larger groups are used within a node of the tree, this leads to larger local optimization problems. However, these larger problems tend to be easier to solve (approximately) than by splitting them into subparts, i.e. there are economies of scale in DSM optimization problems. With an increasing group size, the dynamics within specific households and devices become decreasingly significant. However, unless it is explicitly modelled that this is important, a larger group size may also lead to less locality, i.e. the local balance within the problem is not considered. If the demand at a household level (or small groups in general) is relevant in the objective, then we should use better, more specific scheduling methods that can deal with small groups. One of the possible approaches for such small, local problems may be the exact column generation methods as referenced in Section 3.4.2.4.

One of the main challenges in the optimization approaches that we presented is that the subproblems may be a bit “*too successful*”, especially in the first iterations: if we indicate that the demand should be reduced at a certain time, then all devices might reduce their demand at that time, which introduces a demand gap there, and may introduce a new peak at the time where demand was supposed to be the cheapest. In the algorithms we see that the selective acceptance of new solutions helps to cope with this effect, by which the induced peak demand becomes roughly half as large in each iteration. In addition, it may be useful to spread the optimization requests proactively between devices, as proposed in Section 3.5.5.1.1. A local optimization towards the global objective also helps to reduce this problem.

Even though an optimization based control method may give (near) optimal results according to the objective, this does not always mean that the achieved result is useful from an application perspective. The objective is in general incomplete in the description of what we aim for. Therefore, we may get unanticipated side effects if we formulate the objective in an inappropriate way. For example, if we express in the objective that we want a flat demand profile by e.g. minimizing the distance between the maximum and minimum demand value, then this may motivate the optimization problem to intentionally introduce losses to make the profile more flat. From an application perspective, a flat profile may be good, but this should not come at the cost of a significant increase in total demand.

An important topic for DSM that has received little attention so far is uncertainty. The recommendations give several suggestions to account for uncertainty within the framework of this chapter. We argue, based on the short duration of the planning sessions, that the iterative use of the planning mechanism may already be sufficient for all but the most time critical or repetitive (sub-minute) control tasks, and thereby avoid the pitfalls associated with having uncoordinated planning and control methods. For these control tasks, we believe that the planning should decide on a policy to be used by the control mechanism (which describes how the system responds to observations), and subsequently account for the possible realizations of the use of this policy. In the following chapter we develop a planning method that embraces this view, and is able to cope with uncertainty.

3.7.1. RECOMMENDATIONS

3.7.1.1. RESERVE AND UNCERTAINTY MODELLING

A clear deficiency in the presented approaches is that they do not account for possible deviations from the planned behaviour in the future, except for the “brute force” method of replanning. As discussed in Section 3.4.5.1.4, we should design the approaches such that (some of) the available flexibility remains available. A possible strategy to approach this is a reserve model, comparable to the reserves in a conventional electricity system (see Section 2.3.1.1). The main challenge for DSM in comparison with a conventional reserve model is the storage aspect: most DSM resources have a (near-)fixed energy content, and any reserve that is provided now has to be compensated at a later time, whereas conventional generators “forget” what has been delivered in the past as reserve capacity. Bosman proposes a reserve model that follows the conventional model [40: p. 148], and is thereby limited to conventional generators without storage. He also proposes a static heat capacity reservation strategy for the buffers of microCHPs [40: p. 118], which may however lead to a reduced utilization of the available buffer capacity.

Taking a more mathematical approach, robust optimization may be a good starting point for a deterministic approach to address uncertainty, which ensures that we can address all possible realizations within reasonable bounds (see Section 2.3.1.3.3).

3.7.1.2. STOCHASTIC OPTIMIZATION

Instead of the more constraint oriented approach that we described above, we may also consider a more objective oriented approach to uncertainty with stochastic optimization. The presented model of Section 3.2 may be extended with a scenario dimension to represent multiple possible outcomes, i.e. we obtain patterns with a demand vector $\mathbf{x}_{\mathcal{N},\xi}$ and cost $z_{\mathcal{N},\xi}^{\text{sub}}$ of a node $\mathcal{N}$ for scenarios $\xi \in \Xi$. The resulting problem is a multistage problem, with a corresponding pricing scheme $\mathbf{c}_{\mathcal{N},\xi}$ per scenario. A scenario tree description may be used to represent dependencies between scenarios over time and to reduce the size of the representation. Note that all nodes below a node $\mathcal{N}$ should consider a common scenario set, and that nodes may locally evaluate additional (independent) scenario dimensions.

3.7.1.3. GRID CONSTRAINTS

In this work, we have abstracted from possible internal constraints in the grid. These may in principle be modelled as internal constraints from e.g. the neighbourhood nodes at distribution level. The number of grid related constraints can be very large (e.g. describing the maximum demand or the thermal behaviour of every cable segment in a neighbourhood over time). In the transmission domain, constraint generation techniques (e.g. Benders decomposition, see Section 2.3.2.1.2) are popular to address these type of problems, exploiting that often only a few constraints are actually relevant for the optimal solution. In essence, these approaches first solve the problem without considering most of the constraints of the problem, and iteratively add only the (most) violated constraints to the problem, until all constraints are satisfied.

3.7.1.4. STATISTICAL PATTERN GENERATION

One of the reasons that the planning methods we consider in this chapter need several rounds to make a coordinated planning is that all subproblems try to solve the same problem without considering that the other subproblems address the same problem as well. We may be able to reduce the number of rounds if the subproblems acknowledge this. For example, each subproblem may reason about the expected value of the demand in the parent problem, e.g. turning on a washing machine 30% of the time may give an expected value of $\frac{3}{10}$ times the washing machine demand. On average this may give the desired demand profile. Note that this method should also account for the availability of the other devices. Preliminary experiments with groups of washing machines show that this method can be used to produce an aggregate profile with reasonable accuracy, provided that the distribution of the availability (start times and deadlines) of the other devices is known. It may be possible to bias the problem in the first iterations based on an estimate of the available flexibility of others. We may also consider the randomized pricing scheme of IDDP (Section 3.3.4.2) as a variant of this concept, as it may e.g. increase the prices of 30% of the household population at some times, although Figure 3.8a demonstrates that this approach is not very effective (or at least poorly configured) in the first iterations.

3.7.1.5. COMMUNICATING STRATEGIES

For some DSM problems, dynamic control is not only of interest for the stability of the electricity system (i.e. to globally keep supply and demand in balance at all times), but also of economic interest. For example, PV may on some cloudy and windy days continually alternate between (almost) no production and full production within minutes. In this case, we want a dynamic control strategy where the planning accounts for the switching behaviour, and plans to keep the resources that have low switching costs available. The cheapest available resource is often not in the same household (or neighbourhood). Therefore, we should aim for a communicating control strategy. The steering signals in the planning in this case should also represent how the system makes use of the response capacity, such that we can plan accordingly.

COORDINATION IN SPACE

Abstract

Dynamic dispatch based DSM approaches are robust against disturbances of individual devices, but are generally hard to account for in optimization problems, which makes these approaches less suitable in an economic (cost minimization) context. We present a metaheuristics based method to plan and optimize a system with such a dispatch approach. In this method, an aggregator controls a group of devices through a Walrasian auction mechanism by offering appropriate supply offer curves, which are determined by the planning. The planning uses a simplified supply offer strategy model, and uses metaheuristics to find an assignment for this model. The behaviour of the devices is determined by on-line simulation of the dispatch process. Simulations show that this planning method is robust and competitive with the optimal control in terms of cost (2 – 10% higher cost), also in cases with large forecast errors (30% of daily demand).

supply model 148 · metaheuristics 153 · use case 158 · experiments 172

4.1. INTRODUCTION

CONVENTIONAL scheduling approaches, such as those considered in Chapter 3, may be less appropriate in our world with high uncertainty. If we consider e.g. [19], then we see that accurate device level forecasting is generally difficult. Instead of trying to improve these forecasts, literature suggests to look at the devices from an aggregate perspective, i.e. to treat a group of devices as a single coherent resource. By exploiting statistical averaging between devices (profiting from the law of large numbers), the uncertainty in the aggregate representation decreases. This should make the forecasting of the available flexibility easier, and make the control approach more robust against disturbances from individual devices. The aggregator reasons at a group level, and then translates the group level control decisions back to device decisions at run time. In this way, the aggregator may ignore the details of the individual devices up to a certain extent. For example, for an EV charging DSM problem, we now may estimate that we have to charge 30 out of 100 EVs during the night. In contrast to the earlier approach, we now do not have to guess which 30 are selected (or even worse, if every car has a 30% probability of arriving, we may estimate that none of the cars need to be charged).

☞ Parts of this work were performed as a visiting researcher at the *Flemish Institute for Technological Research (VITO)* in Mol, Belgium.

Run time decisions on the aggregate level (“increase the demand by 50 kW”) have to be translated back to device decisions (“charge car #3, #17, #34, #50, and #53 at 10 kW”). The dispatch mechanism has to take care of the interaction between the aggregate level and the device level. For efficiency and sometimes privacy reasons, the dispatch mechanism may not “micro-manage” the decisions of all devices, but instead use a simplified description of the available control options.

A popular dispatch mechanism for DSM applications is the single commodity double-sided Walrasian auction (see Section 2.3.2.1.5). In this auction, devices have a demand for electricity, and suppliers offer electricity. These bids and offers are described by demand bidding curves and supply offer curves, which describe the relative willingness to pay for electricity (in the case of an electricity consuming device), or to supply electricity at a given selling price (in the case of a supplier). The supply can be e.g. a generator, but usually the supply comes from the grid, i.e. the aggregator buys electricity at the market on behalf of the devices that it controls. The auction determines the price where supply and demand curves meet, and thereby gives the optimal allocation of resources for the time interval for which the auction is responsible. However, if we consider supply or demand that can be shifted over time (which is true in the case of DSM), we get a combinatorial aspect, as the auctions in different time intervals become related. Therefore, we can not just iteratively apply the dispatch mechanism for optimization, but instead we should consider to use some sort of planning approach. For example, the approach in Section 2.3.2.1.4 (Figure 2.1) makes an aggregate model of the available flexibility, determines an aggregate optimal control path according to this model, and then uses a supply offer strategy that follows this path in the auction. Similarly, Molderink [232] (see also Section 2.4.2.2) first schedules all devices to find an aggregate optimal control path, and then follows this path in the auction (without considering the found device-level schedule). In both cases, the “price” in the auction is treated as a steering signal instead of as the actual value of the commodity (electricity) at hand.

However, in both of these examples, we see undesirable behaviour due to a mismatch between the planning model and the dispatch model: the emergent behaviour of the devices that results from the use of the dispatch mechanism may differ from what was predicted by the planning model. Iacovella refers to this mismatch as *dispatch dynamics* [166]. To limit the consequences of the mismatch, the aggregator may account for the dispatch mechanism in the planning. In contrast to Chapter 3, we now first choose the dispatch mechanism, and then try to find a planning mechanism that can cope with it, which can be described as *dispatch centric optimization*. In general, this makes the planning model much more complex, and thereby challenging to optimize.

The classical example of an optimization approach that accounts for the control phase is two-stage stochastic programming, where the “dispatch” is typically determined by simple recourse (i.e. any surplus or shortage of electricity can be bought or sold at some given price). Although this case has been well studied in literature, the expressivity tends to be limited. Instead, we choose to focus on the auction dispatch mechanism. The work in [166] estimates dispatch dynamics with

a MIP model of the auction over time, and uses a small number of “tracer devices” to model the device bidding behaviour. Although this method demonstrates that it is in principle possible to incorporate the dispatch dynamics directly into a mathematical programming model, constructing such a model and keeping it tractable is generally difficult with current MIP solvers, and relies on having a set of devices with fairly homogeneous characteristics to find an accurate representation of the dispatch dynamics with a small number of tracer devices. Alternatively, the work in [265] uses a stochastic DP to determine the optimal bidding strategy of each device. In this case, the auction prices do represent commodity prices, and each device minimizes its own cost against the uncertain auction price outcomes over time. The auction prices are assumed to be independent and normally distributed. However, it is not clear how this local optimization relates to the cost at an aggregate level, unless the aggregate level objective is only determined by the same electricity prices. Summarizing, the existing methods to account for dispatch dynamics are computationally expensive and/or inflexible.

In this chapter, we propose a simulation approach to account for dispatch dynamics of a DSM aggregator in the planning phase of the control. The goal of the DSM aggregator is to optimize some cost function, e.g. the cost to buy electricity for a group of devices. Our main motivation to use an auction dispatch mechanism is to cope with the uncertainty within the devices that the aggregator controls. Hereby, we assume that the aggregator does not have the resources to (completely) mask this uncertainty (e.g. with a generator or a very large battery). Therefore, we do not set up a single aggregate target profile, but instead use a simple supply offer strategy and “measure” the effect of this strategy on the given cost function in a stochastic setting. We use metaheuristic optimization to assign the supply offer strategy, which makes the approach very flexible: we expect it to be able to cope with a wide range of cost functions and devices. Also, exploiting the auction mechanism, the control result is robust against correlated forecast errors.

This chapter is structured as follows. First, we give a structured problem description in Section 4.2, and after some simplifications this leads to a tractable structure for metaheuristic algorithms. We discuss some popular algorithms from this class in Section 4.3. To evaluate the performance of these algorithms, we introduce a small use case with significant uncertainty and a reference optimal solution in Section 4.4. We present the results from simulation experiments with this use case in Section 4.5. Finally, in Section 4.6, we wrap up with conclusions and recommendations for future work.

4.2. PROBLEM STATEMENT

4.2.1. INFORMAL DESCRIPTION

We consider the control problem of a DSM aggregator that manages the exchange of electricity with the grid by controlling a large group of devices. This aggregator controls the behaviour of this group, such that it externally behaves as a unified control-

lable resource with desirable properties. Subsequently, it can offer energy services based on this resource, as the aggregator now is able to determine the distribution of electricity exchange with the grid over time by his choice of control actions.

The energy service that the aggregator offers may be represented by an objective function (e.g. the cost of supply). The goal of the aggregator is to minimize this objective function. The objective usually includes an assessment of the aggregate electricity exchange over a longer time period, for example according to electricity prices over time, or the highest absolute exchange value (which also accounts for possible electricity feed-in peaks).

During execution, the aggregator determines the actions of every device in the group in response to observations; we call this *control*. To effectively manage the electricity exchange over time, the aggregator predicts the behaviour of the system as a consequence of its decisions, and adapt the decisions accordingly; we call this *planning*. Therefore, we speak of a planning phase and a control phase. The planning phase gives information to the control phase, which helps it to reach the aggregator's goal. Although we present planning as a distinct phase, the planning may also work in parallel to the control.

In the setting considered in this section, the aggregator controls the devices through a double sided auction mechanism, and therefore does not need detailed direct information from these devices. The devices bid for electricity on the demand side of the auction, and the aggregator offers electricity on the supply side of the auction. The aggregator is represented in the auction by an economic agent. One of the main contributions of this chapter is a planning approach for this setting.

In the context of this section, the outcome of the control of the aggregator is either a value that specifies the requested electricity exchange in a certain time period (which, over time, gives the requested exchange profile of the aggregator), or a set with multiple supply curve offers that specify supply priorities for a specific time period. The latter occurs especially if the aggregator itself participates in a market, and leads to a vector of supply offer curves for a sequence of time periods.

4.2.1.1. DEMAND SIDE

First, we assume that all devices that are controlled by the aggregator have a demand for electricity. Each of these devices gives a demand bidding curve to the auction mechanism. The bidding curves specify the current possible electricity consumption values for each device, according to the “urgency” to consume electricity. Literature commonly refers to this urgency as the *price*; however, to avoid confusion with monetary prices in the objective (which describe commodity cost rates), we prefer to use the term *priority*. A higher priority (/price) leads to a lower or equal electricity demand. A bidding curve may be read from right to left: only a small part of the demand has a high priority and must absolutely be served, and we can afford to serve demand with a lower priority only when sufficient supply is available (in the context of DSM, this means that the demand is postponed, gradually making it more urgent if we continue to postpone it). More precisely, the shape of the bidding curve somehow reflects the current state of the device, and determines

the available control range. The state of a device depends both on environmental factors (e.g. heat demand) and on the history of electricity exchange. Devices update their bidding curves continually (at irregular intervals), but these changes are often very gradual. For now, we assume that (agents of) devices bid honestly, and do not try to gain advantage by hiding or misrepresenting control options. To keep the model simple, we assume that all devices bid on the demand side of the auction, and support devices that can supply electricity (e.g. PV panels or CHPs) by allowing devices to specify feed-in with negative demand values.

4.2.1.2. AGGREGATION

The auction mechanism, which is managed by the aggregator, combines the device bidding curves into an aggregate demand bidding curve. The aggregate bidding curve has the same structure as a device bidding curve, and couples the possible decisions of all devices, according to their priority to consume electricity. Visually, the bidding curve aggregation can be interpreted as the stacking of bidding curves, as we show in Figure 4.1. The shape of the aggregate demand bidding curve represents the combined state of the device group. Also, this curve gives the range of electricity exchange quantities from which the (economic agent of the) aggregator can choose. Each priority level gives a specific corresponding behaviour for all devices, in accordance to the original device bids. The auction mechanism now has to determine a priority level as the outcome of the auction, which is called the clearing priority.

4.2.1.3. SUPPLY SIDE

To determine the clearing priority, next to the (aggregate) demand curve, the auction mechanism also needs a supply curve. The supply curve is the counterpart of the demand curve: in the supply curve, a higher priority (/price) gives a higher or the same electricity supply (if the demand is more urgent, then the supply will work harder to satisfy the demand). We give an example of a supply curve in Figure 4.2a. Just as for the demand side, the supply side can represent net feed-in with negative values, although in this case the negative values appear for low priority levels (instead of for high priority levels as for the demand side) as the supply function is nondecreasing. That is, we have net feed-in of electricity when there is no (willing) demand to make use of it.

4.2.1.4. AUCTION CLEARING

The auction mechanism compares the supply curve (which goes up with priority) with the demand curve (which goes down with priority), and sets the clearing priority where the supply curve and the demand curve meet. If the curves meet in a unique priority, then this point also gives the equilibrium value, i.e. the quantity of exchanged electricity. If the curves meet in a segment where both curves are flat, then we may pick an arbitrary priority point in the range where supply and demand are equal. In the case where the curves never meet, we choose the

FIGURE 4.1: Demand curve aggregation with two devices.

FIGURE 4.2: Supply curve of the economic agent, and clearing between this supply curve and the demand curve of Figure 4.1c.

value on the demand curve where the curves are closest as the exchange value (we define this more precisely in Section 4.2.2.4). We choose to use the value on the demand curve as the exchange value, as this corresponds with the realized demand from the devices (as declared by the device agents). Figure 4.2b gives an example of the clearing between the aggregate demand curve of Figure 4.1c and the supply curve of Figure 4.2a.

4.2.1.5. OBJECTIVE

The aggregator uses an economic agent to provide the supply curve. Just as devices update their bidding curves, this economic agent may determine the supply curve dynamically. By controlling the shape of the supply curve, the aggregator can control the exchange of electricity with the grid, within the control range offered by the (aggregate demand curve of the) devices.

The economic agent is able to choose either a specific clearing priority or a specific demand quantity (within the control range of the devices) by using very specific supply curves (see Figure 4.3a and Figure 4.3b, respectively). The more general approach is that he gives a “proper” supply curve where the market clearing determines the equilibrium priority and quantity, according to where supply meets demand (Figure 4.3c).

We assume that the economic agent is the only agent at the supply side of the market. However, as we allow negative demand, we can move all other supply sources to the demand side with a sign change (positive supply is negative demand), and therefore this assumption is not a restriction.¹

4.2.1.6. OPTIMIZATION

The aggregator controls the import of electricity, because it wants to minimize its total cost. Whereas minimizing instantaneous cost is often easy (“stop doing everything”), the economic agent also needs to account for the future to minimize overall cost. After all, the decision to consume less now in general leads to more demand somewhere in the future, and the other way around. In particular, the agent may later be forced to consume electricity at an inconvenient time due to device or user constraints. To minimize cost, the economic agent needs to give suitable supply curve offers, by which it manages the exchange of electricity over time. We want to consider the distribution of demand over the day, and may want to consider the day-ahead market. Therefore, we typically consider the cost over a time horizon of 24 hours. As it is for most devices not allowed to postpone demand indefinitely, the demand options in the future depend on the past and on the current demand. We call the method by which the economic agent determines these offers a supply curve offer *strategy*. The problem that we now have is to determine a

1. In fact, even the definition of the economic agent as a supply side agent is a matter of convention. Our auction mechanism implementation internally only supports demand side bids to simplify the implementation. The supply side specifies demand bids with a negative sign. The clearing priority corresponds with the zero crossing of the aggregate demand curve.

(a) Forced clearing priority (to a high priority level, suppressing most non-essential loads).

(b) Forced exchange quantity (2500 W, which here satisfies only urgent loads).

(c) “Proper” supply curve: priority and exchange quantity both depend on clearing, i.e. the point where the supply curve meets the demand curve.

FIGURE 4.3: Example supply curve shapes, together with an example aggregated demand curve and the corresponding clearing outcome. Although the presented supply curves have very different shapes, the clearing priority outcome is the same, giving the same device control actions and exchange quantity.

strategy to offer supply curves over time that minimizes overall cost, for a given cost function. In principle, the economic agent has to specify the strategy up-front, such that we can determine the cost to use this strategy.

4.2.1.7. DEMAND/COST EVALUATION MODEL

A problem that remains is that we are in principle not able to determine the cost up-front, because the device bidding functions are not known until the devices determine their bidding curves over time. However, using a model of the aggregate demand, we can estimate the response of the group of devices to a supply offer

strategy. This gives an estimate of the exchange values over time, which we can use to estimate the cost to use a given supply offer strategy. This model of the aggregate demand may be constructed based on historical data.

To build such a model, we use a bottom-up approach where we rely on additional knowledge on the structure of the underlying devices. For every device, we make a model that describes the bidding behaviour and the internal state change corresponding to the outcome of the auction. By aggregating these device models, we obtain a model for the complete demand side. This aggregation is analog to the bottom-up optimization structure of TRIANA as we describe in Chapter 3, but for now considers only two levels (aggregate level and device level).

As is known from literature, aggregate device bidding functions are hard to capture in optimization problems, because we have to account for the bidding strategies of devices. These bidding strategies often use operations such as conditionals (e.g. if the buffer is near empty, then offer a less efficient control option) and divisions (e.g. a hyperbolic bid using $1/\text{soc}$), which are hard to express or expensive to consider in mathematical programming formulations (e.g. MIPs). Instead, we do not embed the bidding strategy directly, but rely on simulations to determine the possible responses of the devices to a supply curve offer strategy. Typically, simulation models are easier to build than optimization models, and may account for more details of the system at hand. A disadvantage of this approach is that the structure in the problem becomes implicit, which limits the ability to find (exact) analytical results.

In a decentralized setting, a day ahead simulation of a bidding process (with many bidding function changes and subsequent priority changes) can have a significant cost in bandwidth and/or latency, because the devices have to send every bidding function change and the aggregator has to publish every clearing priority change. These operations work in lock-step: the device needs to know the clearing priority to update its state and thereby the next bidding curve, which may again lead to a new clearing priority (this requirement may be relaxed for performance reasons, at the cost of accuracy). In the following discussion, we avoid these costs with a centralized simulation model, where devices submit their simulation model to the aggregator. In practice, this could e.g. be realized using the EFI abstraction to communicate the available control flexibility (Chapter 5). Other alternatives to reduce these costs are discussed in Section 4.6.1.

4.2.2. FORMAL MODEL

In the following, we give a more formal description of the considered problem.

4.2.2.1. DEMAND SIDE

Let $\mathcal{I}^d$ be the set of controllable devices that represent all demand, and let $\mathcal{T} = \{1, \dots, n_t\}$ be a discretization of the time horizon in n_t time intervals, all of length τ (e.g. $\tau = 15 \cdot 60 = 900$ s and $n_t = 96$ for a 24 h horizon).² Each device³ $i \in \mathcal{I}^d$ has

2. While we present the approach with a fixed time interval length of τ , the approach may just as well be implemented with a variable time interval length with minor changes.

3. Although $\mathcal{I}^d$ in principle contains *demand sources*, we simplify the discussion by assuming that all demand sources are controllable devices, and the economic agent is the only source of supply.

a demand bidding curve function $x_{i,t}^d(p)$, which gives the electricity demand of device $i \in \mathcal{I}^d$ for a priority $p \in \mathbb{R}$ in time interval $t \in \mathcal{T}$. As mentioned, these device demand bidding curves reflect the state of each device $i \in \mathcal{I}^d$ in $t \in \mathcal{T}$. Demand bidding curves are by definition monotonically nonincreasing in p :

$$x_{i,t}^d(p_a) \geq x_{i,t}^d(p_b) \quad \forall p_a, p_b : p_a \leq p_b, i \in \mathcal{I}^d, t \in \mathcal{T}, \quad (4.1)$$

and may also represent feed-in by using negative demand values:

$$x_{i,t}^d(p) \in \mathbb{R} \quad \forall i \in \mathcal{I}^d, t \in \mathcal{T}, p \in \mathbb{R}. \quad (4.2)$$

The concrete structure of the bidding curve is device (type) specific, and usually depends on the state of the device at time $t \in \mathcal{T}$.

We assume that the “active” domain and range of each bidding curve are both finite (or can be truncated), i.e. before some priority level $p_{i,t}^{d,\min}$ and after $p_{i,t}^{d,\max}$, the bidding curve is constant:

$$p_{i,t}^{d,\min} = \arg \max_{p'} \forall p \leq p' : x_{i,t}^d(p) = x_{i,t}^d(p') \quad \forall i \in \mathcal{I}^d, t \in \mathcal{T} \quad (4.3)$$

$$p_{i,t}^{d,\max} = \arg \min_{p'} \forall p \geq p' : x_{i,t}^d(p) = x_{i,t}^d(p') \quad \forall i \in \mathcal{I}^d, t \in \mathcal{T}. \quad (4.4)$$

Devices are indifferent to priority changes outside of the “active” domain at either end, and in general also on flat segments of the bidding curve.

4.2.2.2. AGGREGATION

The aggregator sums up all bidding curves of the devices $i \in \mathcal{I}^d$, which gives the aggregate demand bidding curve $x_t^d(p)$:

$$x_t^d(p) = \sum_{i \in \mathcal{I}^d} x_{i,t}^d(p) \quad \forall t \in \mathcal{T}, p \in \mathbb{R}. \quad (4.5)$$

This aggregate bidding curve reflects the state of the whole device group $\mathcal{I}^d$.

The “active” domain of the aggregate bidding curve $[p_t^{d,\min}, p_t^{d,\max}]$ can be defined directly as a variant of (4.3) and (4.4) (replacing $x_{i,t}^d$ with x_t^d), or indirectly as:

$$p_t^{d,\min} = \min_{i \in \mathcal{I}^d} p_{i,t}^{d,\min} \quad \forall t \in \mathcal{T} \quad (4.6)$$

$$p_t^{d,\max} = \max_{i \in \mathcal{I}^d} p_{i,t}^{d,\max} \quad \forall t \in \mathcal{T} \quad (4.7)$$

4.2.2.3. SUPPLY SIDE

Subsequently, let i^s be the economic agent, which is the only source of supply in the auction:

$$\mathcal{I}^s = \{i^s\}. \quad (4.8)$$

The supply source $i \in \mathcal{I}^s (= i^s)$ has a supply offer curve function $x_{i,t}^s(p)$, which gives the electricity supply for a priority $p \in \mathbb{R}$ at time $t \in \mathcal{T}$. Supply offer curves are monotonically nondecreasing in p :

$$x_{i,t}^s(p_a) \leq x_{i,t}^s(p_b) \quad \forall p_a, p_b \in \mathbb{R} : p_a \leq p_b, i \in \mathcal{I}^s, t \in \mathcal{T}, \quad (4.9)$$

and may represent feed-in with negative supply values. In case of multiple supply sources, we define the aggregate supply curve similar to the demand side as the sum of supply curves:

$$x_t^s(p) = \sum_{i \in \mathcal{I}^s} x_{i,t}^s(p) \quad \forall t \in \mathcal{T}, p \in \mathbb{R}. \quad (4.10)$$

In our case with a single supply source i^s , this means:

$$x_t^s(p) = x_{i^s,t}(p) \quad \forall t \in \mathcal{T}, p \in \mathbb{R}. \quad (4.11)$$

At each $t \in \mathcal{T}$, the economic agent of the aggregator chooses a supply offer curve $x_{i^s,t}^s(p)$, and gives this curve to the auction mechanism. The agent may determine the offer curve dynamically at the time of use. Analogous to the demand bidding curves, we assume that the “active” domain and range of the supply offer curves are finite, and denote the “active” domain for the economic agent supply offer curve as $[p_{i^s,t}^{s,\min}, p_{i^s,t}^{s,\max}]$ and for the supply offer curve as $[p_t^{s,\min}, p_t^{s,\max}]$.

As discussed in Section 4.2.1.5, the economic agent may use the supply curve to choose either a specific clearing priority p_t^* (with a step from the lowest to the highest possible supply value at $p = p_t^*$, as in Figure 4.3a) or a specific supply value x_t^s (using a constant bid, as in Figure 4.3b). However, due to possible errors in the “planning” (e.g. demand forecast errors), we recommend against both of these approaches: the first can introduce arbitrarily large perturbations in demand, and the second is not robust under changes in demand volume, unless the supply offer value is adapted frequently. Instead, we believe that the economic agent should make a “proper” supply offer curve that can adapt to changing circumstances, but also does not adapt too abruptly (as in Figure 4.3c). In principle, the economic agent can directly observe $x_t^d(p)$ before it makes an offer, but we do not use this ability.

Next, the auction will (try to) clear the “market”, using the aggregate demand curve and the aggregate supply curve.

4.2.2.4. AUCTION CLEARING

The auction mechanism overlays the (“aggregate”) supply curve of the economic agent with the aggregate demand curve of the device group, and determines the point where supply meets demand.

In some cases, supply and demand never meet. For these cases, we take the demand curve as the “leading” curve, and use a modified supply curve $\tilde{x}_t^s(p)$ to give a meaningful outcome. In a grid-tied context, we can treat the supply curve $x_t^s(p)$ as a control signal, and let the clearing demand $x_t^d(p_t^*)$ give the actual level of supply and demand for a clearing priority p_t^* (as estimated by the device bidding

FIGURE 4.4: Supply and demand may never meet. We correct the supply curve to ensure that we always have an appropriate clearing value.

functions). We identify two cases where supply and demand never meet exactly. First, the range of the supply curve and the demand curve may have no overlap (Figure 4.4a). In this case, we insert the nearest demand level in the supply curve. Second, if the demand or supply curves have discontinuities (e.g. devices with on/off control options), then a curve can “jump” over the other curve (Figure 4.4b). In this case, we copy a bid value from the demand curve to the supply curve. The point where the demand curve and the modified supply curve meet gives the equilibrium priority p_t^* and the equilibrium quantity $\tilde{x}_t^s(p_t^*) = x_t^d(p_t^*)$. Hereby, the electricity exchange with the grid always follows from the aggregate demand outcome from the auction $x_t^d(p_t^*)$, for $t \in \mathcal{T}$, and therefore we can use these terms interchangeably.

The auction clearing is defined as follows. Given the modified supply curve $\tilde{x}_t^s(p)$ and the demand curve $x_t^d(p)$ at time $t \in \mathcal{T}$, we look for clearing priorities $p_t^* \in \mathbb{R}$ such that:

$$\tilde{x}_t^s(p_t^*) = x_t^d(p_t^*) \quad \forall t \in \mathcal{T}. \quad (4.12)$$

Note that finding this priority (or determining whether it exists) is straightforward, as $x_t^d(p)$ is nonincreasing and $\tilde{x}_t^s(p)$ is nondecreasing. To address the “jumps” case, we first define a “forced clearing” priority p_t^* , which is only defined (and equal to p_t^*) if the ranges $[x_t^s(p^{s,\min}), x_t^s(p^{s,\max})]$ and $[x_t^d(p^{d,\max}), x_t^d(p^{d,\min})]$ overlap, with $\epsilon \rightarrow 0$:

$$p_t^* = \begin{cases} \min p : x_t^d(p) \leq x_t^s(p + \epsilon) & \text{if } x_t^d(p^{d,\min}) \geq x_t^s(p^{s,\min}) \text{ and} \\ & x_t^d(p^{d,\max}) \leq x_t^s(p^{s,\max}) \\ \text{undefined} & \text{otherwise} \end{cases} \quad \forall t \in \mathcal{T} \quad (4.13)$$

Now, we define $\tilde{x}_t^s(p)$ such that p_t^* always exists.

$$\tilde{x}_t^s(p) = \begin{cases} x_t^d(p_t^{d,\min}) & \text{if } p \leq p_t^{s,\min} \text{ and } x_t^s(p_t^{s,\min}) > x_t^d(p_t^{d,\min}) \\ x_t^d(p_t^{d,\max}) & \text{if } p \geq p_t^{s,\max} \text{ and } x_t^s(p_t^{s,\max}) < x_t^d(p_t^{d,\max}) \\ x_t^s(p) & \text{otherwise, if } p_t^* \text{ is undefined} \\ \min(x_t^s(p), x_t^d(p_t^*)) & \text{otherwise, if } p \leq p_t^* \\ \max(x_t^s(p), x_t^d(p_t^*)) & \text{otherwise, if } p > p_t^* \end{cases} \quad \forall t \in \mathcal{T}, p \in \mathbb{R} \quad (4.14)$$

In this equation, the first case applies when the supply curve is always higher than the demand curve, the second case applies when the supply curve is always lower than the supply curve, the third case supplements the first two cases outside of the range of interest, and the fourth and fifth cases apply both when there is a relevant “jump” and when supply meets demand exactly.

Based on the behaviour of the auction mechanism, we now evaluate the result of the dispatch process.

4.2.2.5. OBJECTIVE

The dispatch process continually (at irregular intervals) uses the auction mechanism to account for updates from the device bidding functions and the supply function of the economic agent (but, for notational clarity, we still assume that all updates occur synchronously at the begin times of time intervals $t \in \mathcal{T}$). At every time $t \in \mathcal{T}$, the market clearing priority p_t^* gives a demand value $x_t^d(p_t^*)$, which we abbreviate to x_t^d . We denote the electricity exchange profile over time $\langle x_1^d, \dots, x_{n_t}^d \rangle$ as $\mathbf{x}^d$. Similarly, we denote the clearing priority vector $\langle p_1^*, \dots, p_{n_t}^* \rangle$ as $\mathbf{p}^*$. The aggregator has to buy (or sell) the electricity exchange given by $\mathbf{x}^d$ on a higher level electricity market. We denote the costs to buy electricity on these markets with a cost function $z^{\text{ext}}(\mathbf{x}^d)$. The definition of cost is context dependent, but usually follows a common structure. For example, a typical component is the commodity cost $\mathbf{c}$ (per kWh), which usually varies over time. Furthermore, the aggregator may incur internal costs, which we denote as z^{int} . The internal cost may depend both on $\mathbf{x}^d$ and on internal properties of devices (we leave these dependencies implicit when we refer to z^{int}). The aggregator has no direct control over the device internal costs, but relies on the bidding strategy of the devices to limit these costs. We define the total cost $z(\mathbf{x}^d)$ as the sum of internal and external cost: $z(\mathbf{x}^d) = z^{\text{ext}}(\mathbf{x}^d) + z^{\text{int}}$. Note that the cost model is similar to the model of Section 3.2 (where z^{ext} maps to z_x^{up} and z^{int} to z_x^{sub} if we consider the aggregator as a root node x in that model).

We do not impose specific constraints on the structure of these cost functions. However, we do assume that z^{ext} is reasonably smooth in $\mathbf{x}^d$. Although the proposed search methods can cope with a non-smooth cost landscape, the control strategy that we derive from it assumes a cost function that is smooth in $\mathbf{x}^d$ within the relevant range. The same applies to z^{int} . Furthermore, we assume that the device bidding functions try to limit internal cost. For now, we assume that z^{ext} only depends on $\mathbf{x}^d$, and not on any further internal details of the devices.

4.2.2.6. OPTIMIZATION

The goal of the aggregator's economic agent is to minimize $z(\mathbf{x}^d)$. The values of z^{int} and $\mathbf{x}^d$, and thereby also of $z^{\text{ext}}(\mathbf{x}^d)$, depend on the clearing priority outcomes $\mathbf{p}^*$, which depend on the bidding behaviour of the devices and on the supply offers of the economic agent. We call the method by which the economic agent determines its supply offer curves the *supply offer strategy*. We denote the supply offer strategy as π , and the resulting supply curve under π at time $t \in \mathcal{T}$ as $\hat{x}_t^\pi(p)$, with $p \in \mathbb{R}$. In the auction, the clearing with this supply curve gives a clearing priority p_t^π . Furthermore, we denote the resulting demand under π by $\mathbf{x}^d(\pi) = \langle x_1^d(p_t^\pi), \dots, x_{n_t}^d(p_t^\pi) \rangle$, and the objective value by $z(\pi) = z(\mathbf{x}^d(\pi))$. Note that π does not describe prices or dual variables in this context. The definition of the concrete supply curve of the supply offer strategy for time interval t may rely on historic information, including the aggregate demand bidding functions at $\{1, \dots, t\}$ and forecast models derived from this information. For notation clarity, we omit the indexing by π if the policy at hand is clear.

The problem of setting up a supply offer strategy that minimizes $z(\pi)$ is a combinatorial problem: an action in an early time interval affects the available control options (and can indeed incur high costs) in later time intervals. The demand bidding curve $x_t^d(p)$ depends on the previous auction clearing priority outcomes $\langle p_1^\pi, \dots, p_{t-1}^\pi \rangle$, the internal cost may depend on all clearing priority outcomes $\mathbf{p}^\pi = \langle p_1^\pi, \dots, p_{n_t}^\pi \rangle$, and the external cost depends on $\mathbf{x}^d$, which also depends on $\mathbf{p}^\pi$. From these dependencies, it is clear that it is hardly possible to solve this problem exactly, unless we find a specific structure in the problem.

4.2.2.7. DEMAND/COST EVALUATION MODEL

Although our goal is to minimize $z(\pi)$, in general we have incomplete information on the problem at hand, and we can in principle only determine the value of $z(\pi)$ in hindsight once all relevant information is available (or are even unable to determine $z(\pi)$ at all if the information never becomes available). Therefore, we instead work with a cost estimate $\tilde{z}(\pi)$ for which we assume that it is a good approximation of $z(\pi)$, i.e. $\tilde{z}(\pi) \approx z(\pi)$.⁴ As discussed in Section 4.2.1.7, we simulate the auction dispatch process with a bottom-up model to determine $\tilde{z}(\pi)$ for a given value of π . That is, for every time interval $t \in \mathcal{T}$, we gather and aggregate the device bidding curves $x_{i,t}^d(p)$ for $i \in \mathcal{I}^d$, determine the supply curve $\hat{x}_t^{s,\pi}(p)$, clear the market to find the clearing priority p_t^π and the corresponding exchange quantity $x_t^d(p_t^\pi)$, and update the state by applying the control decision corresponding to p_t^π for every device i . At the end, we can use the estimated exchange quantities to determine the estimated cost $\tilde{z}(\pi)$. Note that we do not actually perform the control actions yet, but only apply the control decisions within the simulation model. These operations have a sequential dependency over t : to clear the market at time t , we first need to know the bids of the devices, which depend on their

4. More precisely, as we are only interested in $\tilde{z}(\pi)$ to find a π that minimizes $z(\pi)$, it is also sufficient if $\tilde{z}(\pi)$ gives the same (or a comparable) *ordering* of solutions as $z(\pi)$.

state, which depends on the outcome of the auction at time $t - 1$, which in turn relies on the outcome of the auction at time $t - 2$, and so on, until we arrive at the present (“ $t = 0$ ”).

A major issue in many energy optimization problems, including ours, is the presence of uncertainty. Therefore, minimizing $z(\pi)$ is not possible in general. Instead, we should aim to minimize the expected cost over the optimization horizon $\mathbb{E}\{z(\pi)\}$. In our realization, this means that we minimize the estimate of the expected cost under uncertainty in the planning phase $\mathbb{E}\{\tilde{z}(\pi)\}$. To estimate $\mathbb{E}\{\tilde{z}(\pi)\}$, we use a scenario based (Monte Carlo) uncertainty modelling. More specifically, we introduce a set of possible scenarios $\tilde{\Xi} = \{\tilde{\xi}_1, \dots, \tilde{\xi}_{n_{\tilde{\Xi}}}\}$, which represents (and should be representative of) the uncertainty in the problem. For example, each scenario $\tilde{\xi} \in \tilde{\Xi}$ may define a possible realization of the heat demand $d_{i,t}^{\tilde{\xi}}$ for every device $i \in \mathcal{I}^d$ and $t \in \mathcal{T}$. As the sources of uncertainty in DSM problems are typically highly correlated between devices (e.g. heat demand and PV feed-in), the scenarios should give “global” realizations of the uncertain parameters. To specify the relevance of every scenario $\tilde{\xi} \in \tilde{\Xi}$, we define a probability $p_{\tilde{\xi}}$, which gives the weight of $\tilde{\xi}$ in $\tilde{z}(\pi)$, with $\sum_{\tilde{\xi} \in \tilde{\Xi}} p_{\tilde{\xi}} = 1$. That is, $\tilde{z}(\pi) = \sum_{\tilde{\xi} \in \tilde{\Xi}} p_{\tilde{\xi}} \tilde{z}^{\tilde{\xi}}(\pi)$ ($\approx \mathbb{E}\{z(\pi)\}$), where $\tilde{z}^{\tilde{\xi}}(\pi)$ gives the cost over the optimization horizon of using strategy π in scenario $\tilde{\xi}$. Usually, we make the scenarios equiprobable, i.e. we choose $p_{\tilde{\xi}} = 1/|\tilde{\Xi}|$ for every $\tilde{\xi} \in \tilde{\Xi}$.

In the simulation of the auction, we now have to make separate simulations for every scenario $\tilde{\xi} \in \tilde{\Xi}$. However, these scenarios are independent, and can therefore be considered in parallel. In a decentralized context, we can reduce overhead by combining the transmissions of multiple scenarios (i.e. one large packet instead of $n_{\tilde{\Xi}}$ small packets for each time interval, in both directions). We expect that the scenario set may be quite small, because the dispatch mechanism already provides some robustness. Because of this, the overhead of the multi-scenario approach should remain acceptable.

In the planning, we commit to an assignment of π based on the information from the scenarios $\tilde{\Xi}$. After this, we evaluate the choice of π based on the true expected cost $\mathbb{E}\{z(\pi)\}$. We evaluate $\mathbb{E}\{z(\pi)\}$ analogous to the cost estimation model with a realized scenario set $\Xi = \{\xi_1, \dots, \xi_{n_{\Xi}}\}$. We assume that $\mathbb{E}\{z(\pi)\} = \sum_{\xi \in \Xi} p_{\xi} z^{\xi}(\pi)$, and in the following we choose to write $z(\pi)$ instead of $\mathbb{E}\{z(\pi)\}$ for notational convenience. Note that it is in principle only possible to obtain this information for synthetic scenarios as it requires knowledge on what “could have happened”. We can reuse the auction simulation approach to determine the value of $z(\pi)$ (or approximate it if no information is available at all), using Ξ instead of $\tilde{\Xi}$ as the scenario set.

At this point, we now have a very general optimization model for a DSM problem with an auction dispatch mechanism. In the following, we add more structure to the supply offer strategy, such that we can describe and choose concrete assignments of π .

4.2.3. STRUCTURED SUPPLY CURVE OFFER MODEL

The model presented in Section 4.2.2 still has insufficient structure to be addressed directly by an optimization approach: the supply curves can in principle have an arbitrary (non-decreasing) shape, and may be determined dynamically based on arbitrary knowledge. In the following, we impose further structure to the problem, which leads to a model which is amenable to optimization with a general optimization approach (which we do in Section 4.3).

As a first step, we further restrict the shape of the allowed supply curves to those characterized by a finite set of values. More precisely, we consider n_{ps} values that specify the supply offer values for a given set of n_{ps} priority levels. For example, if we use three values ($n_{ps} = 3$), then the first value may give the supply offer at a low priority level, the second value may give the supply offer at a normal priority level, and the third value may give the supply offer at a high priority level. Between the priority levels for which we specify the supply offer value, we interpolate the supply curve. Below and above the minimum and maximum specified priority level, we respectively use the minimum and maximum specified supply offer value. The amount and distribution of these priority levels gives a tradeoff between accuracy and problem size (and thereby search dimension). Note that if we choose one value ($n_{ps} = 1$), then the supply curves request a specific demand level. However, as discussed before, this approach is not robust to changes in demand.

As a further simplification, we declare the supply curves for all time intervals up-front after the planning, with no further dependencies on the realization. Consequently, we can only use the device bidding curves to react on the internal state of the group. The use of a static supply curve (per time interval) implies that we do not respond to dynamic commodity prices. As a consequence, we can now describe the supply curve offer strategy with an n_t -vector of n_{ps} -tuples (e.g. $96 \cdot 3 = 288$ values).

Note that in general, this supply curve strategy vector is not very variable over time: in a good assignment of these vectors, adjacent tuples are commonly almost the same, due to gradual changes in the pricing scheme and in the aggregate demand curve. Therefore, we may choose to assign the tuples of n_{ps} values for only few time intervals n_{ts} , and then interpolate the remaining values in between. We denote this compact representation as $\hat{\mathbf{x}}$ (of size $n_{ts} \cdot n_{ps}$, e.g. $9 \cdot 3 = 27$ values), and use this representation to derive the supply curves for all t .

Looking back to Section 4.2.2.6, we now find that $\hat{\mathbf{x}}$ completely represents (the parameters of) the supply curve strategy, and therefore we have $\pi \equiv \hat{\mathbf{x}}$ (together with the meaning of these values as described in this section). The goal of the optimization procedure is to find the values $\hat{\mathbf{x}}$ that minimize the objective value $z(\pi)$.

In the following, we continue with a more formal description of this supply curve offer model in Section 4.2.3.1, and give an example of the use of this model in Section 4.2.3.2. Afterwards, in Section 4.3, we propose a method to find the values for the parameters of this model.

4.2.3.1. MODEL

Let $\langle p^s(1), \dots, p^s(n_{p^s}) \rangle$ be the priority levels for which we define the supply offer value. We usually choose $p^s(1) = -1$ and $p^s(n_{p^s}) = 1$, and evenly spread the rest of the values over the range in between, i.e., $p^s(p') = -1 + 2 \frac{p'-1}{n_{p^s}-1}$ for $p' \in \{1, \dots, n_{p^s}\}$. For $n_{p^s} = 3$, this gives the priority levels $p^s(1) = -1$, $p^s(2) = 0$, and $p^s(3) = 1$. By convention, a priority level of -1 corresponds in the device bidding strategies to a (very) low priority level, a level of 0 to a normal priority level, and a level of 1 to a high priority level. The supply offer values for these priority levels change over time. Let $\langle t^s(1), \dots, t^s(n_{t^s}) \rangle$ be the time intervals for which we specify the supply values. We require both the priority levels and the time intervals to be in ascending order:

$$p^s(p') < p^s(p' + 1) \quad \forall p' \in \{1, \dots, n_{p^s} - 1\} \quad (4.15)$$

$$t^s(t') < t^s(t' + 1) \quad \forall t' \in \{1, \dots, n_{t^s} - 1\}. \quad (4.16)$$

Next, we let $\dot{x}_{t'}^s(p')$ specify the supply offer value at time $t^s(t')$ for priority $p^s(p')$. We denote a complete assignment of $\dot{x}_{t'}^s(p')$ for all $t' \in \{1, \dots, n_{t^s}\}$ and $p' \in \{1, \dots, n_{p^s}\}$ as $\dot{\mathbf{x}}$. To construct nondecreasing supply curves, these supply values have to be in nondecreasing order for each t' :

$$\dot{x}_{t'}^s(p') \leq \dot{x}_{t'}^s(p' + 1) \quad \forall t' \in \{1, \dots, n_{t^s}\}, p' \in \{1, \dots, n_{p^s} - 1\}. \quad (4.17)$$

We define an interpolation function $\text{interp}(\langle a_1, \dots, a_n \rangle, \langle b_1, \dots, b_n \rangle, a')$ that linearly interpolates at a' from the known values $\langle a_1 \rightarrow b_1 \rangle, \dots, \langle a_n \rightarrow b_n \rangle$ (similar to `interp1` in MATLAB), and extends the outer values b_1 and b_n at each end:

$$\begin{aligned} \text{interp}(\langle a_1, \dots, a_n \rangle, \langle b_1, \dots, b_n \rangle, a') = & \\ \begin{cases} b_1 & \text{if } a' \leq a_1 \\ b_n & \text{if } a' \geq a_n \\ \frac{a_{j+1}-a'}{a_{j+1}-a_j} b_j + \frac{a'-a_j}{a_{j+1}-a_j} b_{j+1} & \text{if } \exists j \in \{1, \dots, n-1\} : a_j \leq a' \leq a_{j+1} \end{cases} & \\ \forall a' \in \mathbb{R}. & \quad (4.18) \end{aligned}$$

The sequence $\langle a_1, \dots, a_n \rangle$ should be increasing. To determine the supply offer $\dot{x}_{t'}^s(p)$ at $t^s(t')$, we now interpolate the priority p :

$$\dot{x}_{t'}^s(p) = \text{interp}(\langle p^s(1), \dots, p^s(n_{p^s}) \rangle, \langle \dot{x}_{t'}^s(1), \dots, \dot{x}_{t'}^s(n_{p^s}) \rangle, p) \quad \forall t' \in \{1, \dots, n_{t^s}\}, p \in \mathbb{R}. \quad (4.19)$$

Finally, we interpolate $\dot{x}_{t'}^s(p)$ over time to determine the supply offer curve $x_{i^s,t}^s(p)$ of the economic agent (which, in our case, is equal to $x_t^s(p)$):

$$x_{i^s,t}^s(p) = \text{interp}(\langle t^s(1), \dots, t^s(n_{t^s}) \rangle, \langle \dot{x}_1^s(p), \dots, \dot{x}_{n_{t^s}}^s(p) \rangle, t) \quad \forall t \in \mathcal{T}, p \in \mathbb{R} \quad (4.20)$$

It is obvious that the resulting supply offer curve is nondecreasing.

(a) Supply offer curve parameters over time $\hat{\mathbf{x}}^\pi$ with the corresponding supply offer curves $x_{i^s,t}^{s,\pi}(-1)$, $x_{i^s,t}^{s,\pi}(0)$, and $x_{i^s,t}^{s,\pi}(1)$, and an exchange quantity outcome trace $x_t^{d,\xi}(p_t^{\pi,\xi})$ ($= \mathbf{x}^{d,\xi}(\hat{\mathbf{x}}^\pi)$).

(b) Supply curve $x_{i^s,13:00}^{s,\pi}(p)$, with corresponding demand bidding curve of devices $x_{13:00}^{d,\pi,\xi}(p)$.

FIGURE 4.5: Example of supply curve representation over time (left), with the corresponding expanded supply curve at 13:00 (right), with one example exchange quantity outcome trace (left).

4.2.3.2. EXAMPLE

To clarify the restricted supply curve offer model, we give an example in Figure 4.5a. We use the example parameters as mentioned before in this section (9 time points, 3 priority levels). The economic agent first determines an assignment of the supply offer strategy parameters $\hat{\mathbf{x}}$. In the figure, the marks give the values $\langle \langle t_{i^s}(1), \hat{x}_1(p') \rangle, \dots, \langle t_{i^s}(n_{i^s}), \hat{x}_{n_{i^s}}(p') \rangle \rangle$ for the priorities $p \in \{-1, 0, 1\}$ (which are indexed by $p' \in \{1, 2, 3\}$ to make the formal definition in (4.15) easier): $\triangle$ gives the supply offers over time for a low priority level ($p = -1$), $\square$ for a normal priority level ($p = 0$), and ∇ for a high priority level ($p = 1$). To make this relation more clear, in Figure 4.5a we mark the point $\hat{x}_3(1) = x_{i^s,6:00}^{s,\pi}(-1) = 1.1$ kW with $*$. The corresponding lines through these points respectively give $x_{i^s,t}^{s,\pi}(-1)$, $x_{i^s,t}^{s,\pi}(0)$, and $x_{i^s,t}^{s,\pi}(1)$ over time. From these lines, we can derive a supply curve at every time $t \in \mathcal{T}$. For example, at 13:00, we interpolate from the values of $\hat{\mathbf{x}}$ at 12:00 and 15:00, giving the values $x_{i^s,13:00}^{s,\pi}(-1) = 1.8$ kW, $x_{i^s,13:00}^{s,\pi}(0) = 3.5$ kW, and $x_{i^s,13:00}^{s,\pi}(1) = 8.7$ kW. After the planning phase, the supply curve is known for all $t \in \mathcal{T}$.

In Figure 4.5b, we derive the supply curve at $t = 13:00$ from the mentioned lines in Figure 4.5a. The supply curve straightforwardly follows from the projection of their value at $t = 13:00$ onto the corresponding priority values $-1, 0$, and 1 .

FIGURE 4.6: Overview of dispatch centric optimization.

The figure shows this (scenario independent) supply curve together with an example of a corresponding aggregate demand curve from the devices at the same time t . To determine this aggregate demand curve at $t = 13:00$, we have first simulated the clearing process in a scenario $\tilde{\xi}$ until $t = 13:00$ (because the bids depend on the history of the devices). From the intersection of the supply curve with the aggregate demand curve, we can find the clearing priority $p_t^{\pi, \tilde{\xi}}$, which gives the exchange quantity (and thereby the demand value) $x_t^{d, \pi, \tilde{\xi}}(p_t^{\pi, \tilde{\xi}})$. In the figure, we find $p_{13:00}^{\pi, \tilde{\xi}} = 0.14$ and $x_{13:00}^{d, \pi, \tilde{\xi}}(0.14) = 4.2$ kW. By iteratively expanding the supply curve and clearing it with the demand curve, we find an exchange quantity trace $\mathbf{x}^{d, \tilde{\xi}}(\tilde{\mathbf{x}}^\pi)$, i.e. the realized demand over time, which we show in Figure 4.5a as —. Note that this figure implicitly also gives the clearing prices over time (if the demand is within the range of the supply curve): for example, the exchange value $x_{13:00}^{d, \pi, \tilde{\xi}} = 4.2$ kW is at 14% between $x_{i^s, 13:00}^{s, \pi}(0)$ and $x_{i^s, 13:00}^{s, \pi}(1)$, giving the priority value $p_{13:00}^{\pi, \tilde{\xi}} = 0.14$. The clearing priority at t in general influences the aggregate demand curve at subsequent time intervals, which explains the dependency of the demand curve on (part of) π . In principle, the shape of the aggregate demand curve in the future is unknown, but we can speculate on possible realizations of this curve, and thereby of $x_t^{d, \pi, \tilde{\xi}}(p_t^{\pi, \tilde{\xi}})$, by simulation of the devices in scenarios $\tilde{\xi} \in \tilde{\Xi}$ (as discussed in Section 4.2.2.7).

4.2.4. OVERVIEW

Before we proceed with the definition of the optimization procedure in Section 4.3, we give an overview of what we now have, and illustrate this in Figure 4.6.

We first consider the planning phase in Figure 4.6a, which (very) roughly corresponds with the planning phase in TRIANA. The planning phase has to assign the parameters for the supply offer strategy $\hat{\mathbf{x}}^\pi$, which we later use in the control phase. We assume that some mechanism, here labelled “oracle”, chooses these values such that the estimated cost $\tilde{z}(\hat{\mathbf{x}}^\pi)$ is minimized. The marks ($\triangle$, $\square$, and ∇) represent the chosen values for $\hat{\mathbf{x}}^\pi$, and the lines between these marks describe the corresponding supply offer values, as in Figure 4.5a (for space reasons, the figure uses $n_{ts} = 3$). From this assignment, we can determine a supply curve $x_{i^s,t}^s(p)$ for every $t \in \mathcal{T}$, as in Figure 4.5. As in principle we do not have direct knowledge of the future, we consider a set of hypothetical scenarios $\tilde{\Xi} = \{\tilde{\xi}_1, \dots, \tilde{\xi}_{n_{\tilde{\Xi}}}\}$ (each of which may e.g. give a realization of the heat demand for all devices), as explained in Section 4.2.2.7. In TRIANA, determining $\tilde{\Xi}$ is known as the forecasting or prediction phase. For a given assignment of $\hat{\mathbf{x}}^\pi$, we predict what happens if we use this assignment in each of these scenarios: we sequentially (over $t \in \mathcal{T}$) determine the aggregate bidding curve $x_t^{d,\pi,\tilde{\xi}}(p)$, hold an auction to determine the clearing priority $p_t^{\pi,\tilde{\xi}}$ and exchange quantity (which is equal to the demand quantity) $x_t^{d,\tilde{\xi}}(\hat{\mathbf{x}}^\pi)$, and update the state of all devices according to the clearing priority $p_t^{\pi,\tilde{\xi}}$. Note that in our simplified supply offer strategy the supply offer curve $x_{i^s,t}^s(p)$ changes over time, but not between scenarios. The demand bidding curves of the devices depend on their state, which depends both on the scenario at hand and on (the decisions resulting from) the clearing priority outcomes in the past. As we consider multiple scenarios in parallel (with a separate state for each scenario), the plots in the figure present $n_{\tilde{\Xi}} = 3$ scenarios (where applicable). The plot labelled “auction” shows the supply offer curve (increasing line) and the demand bidding curves (decreasing lines, one per scenario), with the corresponding clearing priority and exchange quantity. The plot labelled “outcomes” gives the exchange quantity outcomes over time (one line per scenario), which we denote as $\mathbf{x}^{d,\tilde{\xi}}(\hat{\mathbf{x}}^\pi)$, along with the assignment of $\hat{\mathbf{x}}^\pi$. Based on the outcomes, we can determine the estimated cost $\tilde{z}(\hat{\mathbf{x}}^\pi)$. For example, if the cost is only determined by the (estimated) commodity prices $\tilde{\mathbf{c}}$, then the total estimated cost is $\tilde{z}(\hat{\mathbf{x}}^\pi) = \sum_{\tilde{\xi} \in \tilde{\Xi}} p_{\tilde{\xi}} \tau \tilde{\mathbf{c}}^\top \mathbf{x}^{d,\tilde{\xi}}(\hat{\mathbf{x}}^\pi)$, where τ still gives the time interval length and $p_{\tilde{\xi}}$ the probability weight of scenario $\tilde{\xi}$ (as discussed in Section 4.2.2.7). We now take this estimated cost as representative of the real cost $z(\hat{\mathbf{x}}^\pi)$.

In the “real-time” control phase, illustrated in Figure 4.6b, we apply our supply offer strategy, using the assignment $\hat{\mathbf{x}}^\pi$ that is the result from the planning phase. This phase roughly corresponds with the control phase in TRIANA, but takes a more central role in this chapter. For evaluation purposes, we consider not only a single realization scenario, but evaluate the assignment for a different set of possible scenarios $\Xi (\neq \tilde{\Xi})$. Just as in the planning phase, we now derive supply offer curves $x_{i^s,t}^s(p)$ from $\hat{\mathbf{x}}^\pi$, find the clearing exchange values $\mathbf{x}^{d,\xi}(\hat{\mathbf{x}}^\pi)$ for every $\xi \in \Xi$ and $t \in \mathcal{T}$, and determine the expected real cost $z(\hat{\mathbf{x}}^\pi)$ (over the scenarios in Ξ). This cost function uses the real commodity prices $\mathbf{c}$. For now, as our supply offer strategy does not adapt to dynamic prices, we assume $\tilde{\mathbf{c}} \approx \mathbf{c}$, and thereby our evaluation uses $\tilde{\mathbf{c}} = \mathbf{c}$.

As a reference for evaluation, we also compute an optimal solution for each scenario without considering the dispatch mechanism, and denote the exchange values for these solutions as $\mathbf{x}^{*,\xi}$ for $\xi \in \Xi$, and the weighted optimal objective value as z^* . The value of $z(\hat{\mathbf{x}}^\pi)$ is always greater than (or, rarely, equal to) z^* , as z^* considers the same minimization problem with less restrictive constraints, and has perfect information. We later use $z(\hat{\mathbf{x}}^\pi)/z^* - 100\%$, i.e. the relative distance from z^* , as a metric to assess the quality of assignments of $\hat{\mathbf{x}}^\pi$ (under the assumption $z^* > 0$). This value gives the (upper bound on the) relative distance to the best possible assignment of $\hat{\mathbf{x}}^\pi$. Note that, even if we actually minimize $\tilde{z}(\hat{\mathbf{x}}^\pi)$, a different assignment of $\hat{\mathbf{x}}^\pi$ may still give a lower value for $z(\hat{\mathbf{x}}^\pi)$, depending on how well $\tilde{\Xi}$ represents Ξ with respect to the supply offer and device bidding strategies.

In the following, we propose a mechanism to assign values to the variables $\hat{\mathbf{x}}^\pi$ of the supply offer strategy (i.e. an approximation of the “oracle”).

4.3. OPTIMIZATION WITH METAHEURISTICS

In Section 4.2, we have defined a DSM problem where we have to determine the control policy of an aggregator, for which we assume that devices are controlled through a double sided auction mechanism. In particular, we have to determine the parameters that define the supply offer strategy of the economic agent, which acts on behalf of the aggregator in the auction to control the exchange of electricity. We denote these parameters as $\hat{\mathbf{x}}^\pi$. Our goal is to minimize an objective $z(\hat{\mathbf{x}}^\pi)$.

Since this problem has a complex internal structure, we choose to treat the problem as if it has very little structure. That is, we treat the dispatch mechanism as an observable “black box” with parameters, and a limited knowledge on the behaviour of the system with respect to these parameters. We assume that we are able to simulate the behaviour of the system, including the dispatch mechanism. Using these features, the goal is to find good values for the parameters $\hat{\mathbf{x}}^\pi$.

To find these parameters, we use a general optimization approach based on metaheuristics. Metaheuristics are a popular class of (often closely related) optimization algorithms, addressing a wide range of problems, with very few assumptions on the problem at hand. Many of these algorithms rely on a computational analogy to evolution: a population of solutions “evolves” from earlier candidate solutions, and are subjected to random mutations. For some algorithms, the population contains only a single solution. The user has to provide a fitness function, which determines the quality of a solution. A fitness function may correspond directly with a cost function, except that the sense is negated (a high fitness gives a low cost). These algorithms usually gradually switch between a global, more randomly based search strategy, and a local search strategy, intensifying the search around the best known solutions. We consider three algorithms: random optimization, simulated annealing, and genetic algorithms.

Although metaheuristics can in principle address arbitrary optimization problems, the number of decisions that can be addressed by evolutionary algorithms in reasonable time depends highly on the structure of the problem, and in particular

on how well the local search is able to find improving solutions, and the computational cost of evaluating the fitness function. Metaheuristic algorithms need a suitable representation of the problem (structured according to the assumptions of the algorithm) and parameter tuning to give good results; to compensate this, *hyperheuristics* have been developed as an approach to determine which metaheuristic should be used [50]. Optimality is usually also not guaranteed.

In a smart grid context, it may not be a good choice to consider all device decisions separately as variables. A direct application of a metaheuristic algorithm to a DSM problem from practice may fail, because the number of control variables is very large, and has limited (exposed) structure. A better choice is to make decisions based on a summarized representation, with a small number of decision variables that in turn determine the values of all other control variables. Still, the dispatch mechanism should provide robustness, and perform well on average for the objective of the use case at hand. We have defined such a summarized representation for the problem at hand in Section 4.2.3, and we use this representation when applying the metaheuristic.

In the following, we first give an overview in Section 4.3.1 of the (small) changes needed to apply a general optimization approach to the problem description of Section 4.2.4. Next, we discuss three well-known metaheuristics that we apply to our problem: random optimization (Section 4.3.2), simulated annealing (Section 4.3.3), and a genetic algorithm (Section 4.3.4). We acknowledge that more sophisticated metaheuristic search algorithms are available in literature (see e.g. [35] for a literature review), but leave their evaluation to future work. After this, we continue in Section 4.4 with an example use case that we use in Section 4.5 to evaluate these algorithms.

4.3.1. OVERVIEW

In Figure 4.7, we illustrate how we can incorporate a general optimization approach (i.e. metaheuristic optimization), following the structure of Figure 4.6 (explained in Section 4.2.4). In the planning phase (Figure 4.7a), instead of an “oracle”, we now have a search algorithm that can propose values for $\hat{\mathbf{x}}^\pi$, and use the “black box” to determine the corresponding cost estimate $\tilde{z}(\hat{\mathbf{x}}^\pi)$. Note that we use much less information than what is available at this point (e.g. the aggregate bidding curves, clearing priorities, and exchange values for each scenario). The search algorithm may probe the system at most n_k times to limit the computational and communication cost of the procedure. We denote the assignment of $\hat{\mathbf{x}}^\pi$ in the k^{th} attempt as $\hat{\mathbf{x}}^{(k)}$. This assignment is usually a variant of the most successful assignment until k , i.e. the *selected* assignment after iteration $k - 1$ (denoted by the subscript _{sel}). By probing the system repeatedly, the search algorithm may find a suitable assignment for $\hat{\mathbf{x}}^\pi$. The bottom left plot in the figure shows the evolution of the estimated cost over the iterations. The cost of the selected assignment decreases with k as we discover new assignments with lower cost. Finally, we commit to an assignment $\hat{\mathbf{x}}^\pi = \hat{\mathbf{x}}_{\text{sel}}^{(n_k)}$ for the control phase.

FIGURE 4.7: Overview of metaheuristic optimization approach (derived from Figure 4.6).

The control phase (Figure 4.7b) remains the same as before. In a simulation context, we can determine the expected realized cost in execution $z(\mathbf{x}^{(k)})$ for every $k \in \{0, \dots, n_k\}$, and compare this to $\tilde{z}(\mathbf{x}^{(k)})$ (however, the planning phase can not observe $z(\mathbf{x}^{(k)})$). The bottom right plot in the figure gives $z(\mathbf{x}_{\text{sel}}^{(k)})$ together with $\tilde{z}(\mathbf{x}_{\text{sel}}^{(k)})$, and shows the trend that $z(\mathbf{x}_{\text{sel}}^{(k)})$ tends to go downward together with $\tilde{z}(\mathbf{x}_{\text{sel}}^{(k)})$, even though the actual values are markedly different. This relation shows that the dispatch mechanism can adapt to different circumstances, even though Ξ may differ significantly from $\tilde{\Xi}$.

4.3.2. RANDOM OPTIMIZATION

Random optimization [18] starts from some initial solution $\mathbf{x}_{\text{sel}}^{(0)}$, and subsequently attempts random (small) moves to improve the fitness function. If the attempt is successful (i.e. the fitness value improved), then the current solution is replaced. By repeating this procedure iteratively, the fitness function approaches a local optimum. Random optimization is easy to implement, but is known to be susceptible to getting stuck in local optima.

To apply random optimization, in addition to the fitness function, we need a mutation operator which proposes new candidate solutions. First, in iteration k , we have a current solution $\mathbf{x}_{\text{sel}}^{(k-1)}$. Let $\mathbf{x}^{(k)}$ be the candidate solution that we derive from $\mathbf{x}_{\text{sel}}^{(k-1)}$. Our used mutation operator adds a normally distributed random value with a standard deviation σ to each element of $\mathbf{x}$. We denote the outcome of this

random variable as $d^{(k)}(t', p')$. To make the search more focused, but still allow for larger changes to be considered, we choose to apply each of these additions with a probability p_+ , and let $b^{(k)}(t', p')$ indicate the outcome of this choice. We find that this conditional addition reduces the number of needed iterations, and thereby speeds up the search significantly. Finally, we ensure that constraint (4.17) remains satisfied by “pushing down” the values for the lower priorities with the values for the higher priorities (recursively referring to the exchange value $\tilde{x}_{t'}^{(k)}(p'+1)$ for the next priority level). This gives the following definition for the mutation operation:

$$\tilde{x}_{t'}^{(k)}(p') = \min \begin{cases} \tilde{x}_{\text{sel}, t'}^{(k-1)}(p') + d^{(k)}(t', p') \cdot b^{(k)}(t', p'), & \forall k \in \{1, \dots, n_k\}, \\ \tilde{x}_{t'}^{(k)}(p'+1) & \text{if } p' < n_{p^s}, \\ \infty & \text{if } p' = n_{p^s} \end{cases} \quad \begin{matrix} t' \in \{1, \dots, n_{t^s}\}, \\ p' \in \{1, \dots, n_{p^s}\}, \end{matrix} \quad (4.21)$$

with

$$d^{(k)}(t', p') \sim \mathcal{N}(0, \sigma^2) \quad \forall k \in \{1, \dots, n_k\}, t' \in \{1, \dots, n_{t^s}\}, p' \in \{1, \dots, n_{p^s}\} \quad (4.22)$$

$$b^{(k)}(t', p') \sim B(1, p_+) \quad \forall k \in \{1, \dots, n_k\}, t' \in \{1, \dots, n_{t^s}\}, p' \in \{1, \dots, n_{p^s}\}, \quad (4.23)$$

where $\mathcal{N}(\mu, \sigma^2)$ samples a random number from a normal distribution with mean μ and standard deviation σ , and $B(1, p)$ gives the value 1 with a probability p , and 0 otherwise (i.e. the Bernoulli distribution). Next, we only accept the change (select a different assignment) if it has a lower cost:

$$\tilde{\mathbf{x}}_{\text{sel}}^{(k)} = \begin{cases} \tilde{\mathbf{x}}^{(k)} & \text{if } \tilde{z}(\tilde{\mathbf{x}}^{(k)}) < \tilde{z}(\tilde{\mathbf{x}}_{\text{sel}}^{(k-1)}) \\ \tilde{\mathbf{x}}_{\text{sel}}^{(k-1)} & \text{if } \tilde{z}(\tilde{\mathbf{x}}^{(k)}) \geq \tilde{z}(\tilde{\mathbf{x}}_{\text{sel}}^{(k-1)}). \end{cases} \quad \forall k \in \{1, \dots, n_k\}. \quad (4.24)$$

4.3.3. SIMULATED ANNEALING

Simulated annealing is a variant of random optimization that aims to overcome local optima by randomly accepting solutions that are worse than the current solution. In this way, a larger part of the state space is explored. The method gradually decreases the probability of accepting a suboptimal solution. By this, we still approach a local optimum, yet the random “excursions” decrease the risk of reaching a point that does not correspond with the global optimum.

In principle, simulated annealing is similar to the method described in Section 4.3.2. However, instead of (4.24) we use a different acceptance criterion. We select non-improving solutions with a probability $p_{\text{accept}}^{(k)}$:

$$p_{\text{accept}}^{(k)} = e^{-\tilde{c}_{\text{accept}} \frac{\tilde{z}(\tilde{\mathbf{x}}^{(k)}) / \tilde{z}(\tilde{\mathbf{x}}_{\text{sel}}^{(k-1)})}{1 - k/n_k + \epsilon}} \quad \forall k \in \{1, \dots, n_k\}, \quad (4.25)$$

and record the decision in $\mathbf{d}^{(k)}$:

$$\mathbf{d}^{(k)} \sim B(1, p_{\text{accept}}^{(k)}) \quad \forall k \in \{1, \dots, n_k\}. \quad (4.26)$$

This $p_{\text{accept}}^{(k)}$ depends on the quality of the solution (assignments with a low cost are more promising than assignments with a high cost), and on the distance to n_k (if we have enough iterations left, then we can still afford to diverge). Furthermore, the parameter $c_{\text{accept}} > 0$ can be used for tuning. The acceptance criterion now becomes:

$$\mathbf{x}_{\text{sel}}^{(k)} = \begin{cases} \mathbf{x}^{(k)} & \text{if } \tilde{z}(\mathbf{x}^{(k)}) < \tilde{z}(\mathbf{x}_{\text{sel}}^{(k-1)}) \text{ or } d^{(k)} = 1 \\ \mathbf{x}_{\text{sel}}^{(k-1)} & \text{otherwise} \end{cases} \quad \forall k \in \{1, \dots, n_k\}. \quad (4.27)$$

4.3.4. GENETIC ALGORITHM

Genetic algorithms work on the presumption that good solutions are structurally related, and that we can therefore derive good candidate solutions by combining parts of multiple earlier solutions. Where the search approaches in Section 4.3.2 and Section 4.3.3 remember only a single solution, genetic algorithms maintain a set (“population”) of promising solutions. New candidate solution are made by combining members from this population. To overcome local minima (“inbreeding”), genetic algorithms introduce random mutations to the new candidate solution. Many variants of combination and mutation operations exist, which e.g. “cut and paste” parts of the assignment vectors from population members, take linear combinations, shift values, and add random values. Which operations are most effective is problem specific, and depends on the structure in the problem.

We use the following variant, which can be seen as a quite restricted genetic algorithm. The reason for the restriction is the high computational effort to evaluate a single solution. Let $\mathcal{X}_{\text{is}}^{(k)}$ be the population after iteration k , which contains (no more than) the $n_{\mathcal{X}}$ most promising solutions, with the lowest estimated cost. We start with an initial population $\mathcal{X}_{\text{is}}^{(0)}$ with a single solution $\{\mathbf{x}_{\text{sel}}^{(0)}\}$ that takes the same value as for the earlier presented algorithms. In every iteration, we determine one new offspring of the population. For this, we randomly select a set $\mathcal{X}_{\text{gen}}^{(k)}$ of size $n_{\mathcal{X},\text{gen}}$ from the population $\mathcal{X}_{\text{is}}^{(k-1)}$, assign random normalized weights $\mathbf{w}_{\mathcal{X},\text{gen}}^{(k)}$ to these solutions, and take a linear combination of these solutions, i.e. we determine a base solution $\mathbf{x}_{\text{gen}}^{(k)}$, which is defined as:

$$\mathbf{x}_{\text{gen},t'}^{(k)}(p') = \left(\mathbf{w}_{\mathcal{X},\text{gen}}^{(k)} \right)^\top \left\langle \mathbf{x}_{t'}(p') \mid \mathbf{x} \in \mathcal{X}_{\text{gen}}^{(k)} \right\rangle \quad (4.28)$$

$$\forall k \in \{1, \dots, n_k\}, t' \in \{1, \dots, n_{t^s}\}, p' \in \{1, \dots, n_{p^s}\},$$

with

$$\mathcal{X}_{\text{gen}}^{(k)} = \left\langle \text{choice}(\mathcal{X}_{\text{is}}^{(k-1)}) \mid j \in \{1, \dots, n_{\mathcal{X},\text{gen}}\} \right\rangle \quad \forall k \in \{1, \dots, n_k\} \quad (4.29)$$

$$\dot{w}_{\mathcal{X},\text{gen}}^{(k)}(j) = \mathcal{U}(0,1) + \epsilon \quad \forall j \in \{1, \dots, n_{\mathcal{X},\text{gen}}\}, k \in \{1, \dots, n_k\} \quad (4.30)$$

$$w_{\mathcal{X},\text{gen}}^{(k)}(j) = \frac{\dot{w}_{\mathcal{X},\text{gen}}^{(k)}(j)}{\mathbf{1}^\top \mathbf{w}_{\mathcal{X},\text{gen}}^{(k)}} \quad \forall j \in \{1, \dots, n_{\mathcal{X},\text{gen}}\}, k \in \{1, \dots, n_k\}, \quad (4.31)$$

where $\text{choice}(X)$ selects one member from a set X with uniform probability. After this, we apply the mutation operator from (4.21), replacing $x_{\text{sel}, t'}^{(k-1)}$ with $x_{\text{gen}, t'}^{(k)}$, which gives us the offspring solution $\mathbf{x}^{(k)}$. For this solution, we determine the estimated objective value $\tilde{z}(\mathbf{x}^{(k)})$, and update the population by adding the offspring solution to the population. This updating implies that the worst solution of the new population is deleted, i.e. we have to check if $\mathbf{x}^{(k)}$ is better than the worst solution in $\mathcal{X}_{is}^{(k-1)}$:

$$\mathbf{x}_{\mathcal{X}, \max}^{(k)} = \arg \max_{\mathbf{x} \in \mathcal{X}_{is}^{(k)}} \tilde{z}(\mathbf{x}) \quad \forall k \in \{0, \dots, n_k\} \quad (4.32)$$

$$\mathcal{X}^{(k)} = \begin{cases} \mathcal{X}^{(k-1)} \cup \{\mathbf{x}^{(k)}\} & \text{if } |\mathcal{X}^{(k-1)}| < n_{\mathcal{X}} \\ \mathcal{X}^{(k-1)} \cup \{\mathbf{x}^{(k)}\} \setminus \{\mathbf{x}_{\mathcal{X}, \max}^{(k-1)}\} & \text{if } \tilde{z}(\mathbf{x}^{(k)}) < \tilde{z}(\mathbf{x}_{\mathcal{X}, \max}^{(k-1)}), \\ & |\mathcal{X}^{(k-1)}| = n_{\mathcal{X}}, \text{ and } \forall k \in \{1, \dots, n_k\} \\ & \mathbf{x}^{(k)} \notin \mathcal{X}^{(k-1)} \\ \mathcal{X}^{(k-1)} & \text{otherwise,} \end{cases} \quad (4.33)$$

and we select the best solution from $\mathcal{X}^{(k)}$:

$$\mathbf{x}_{\text{sel}}^{(k)} = \mathbf{x}_{\mathcal{X}, \min}^{(k)} = \arg \min_{\mathbf{x} \in \mathcal{X}_{is}^{(k)}} \tilde{z}(\mathbf{x}). \quad \forall k \in \{0, \dots, n_k\}. \quad (4.34)$$

This concludes the description of the variant of genetic algorithms that we consider. The main difference with a classical genetic algorithm is that the generation of offspring is performed sequentially, and always when a better solution is found it is directly added to the population. By this sequential approach, the new offspring can immediately contribute to the generation of the following offspring.

In the following, we present an example use case by which we evaluate our control approach and the corresponding optimization algorithms that we presented here. We continue with simulation experiments on this use case in Section 4.5.

4.4. EXAMPLE USE CASE WITH HEAT PUMPS

To evaluate the control approach and to demonstrate to what extent the approach can match the objective performance of a “conventional” optimization based control strategy, we introduce an example use case. We take a group of 10 heat pumps from the Flex Street scenario (Appendix B). Each heat pump has an operating range of 0 – 2000 W_{el}, a *coefficient of performance* (COP) of 5, and a 8 kWh heat buffer. The buffer is lossless (or the loss is assumed to be accounted for in the demand). An auxiliary resistance heating rod is available for each buffer as well

(with a “COP” of 1), which can be used to meet heat demand in exceptional circumstances or when the buffer turns out not to have been charged enough in advance. We assume that the heat pump and the heating rod are together always able to serve the heat demand directly, and that we are not allowed to shift or curtail the heat demand itself (without the buffer). Note that the use of the heating rod is very expensive due to its low efficiency, which means that its use should in general be avoided (the heating efficiency of the heating rod is 5 times lower than of the heat pump). The objective is to fulfil the heat demand at minimum cost, where cost is defined as electricity cost on the market, plus a (small) quadratic balancing term which avoids excessive exchange peaks during periods with low prices. We choose the balancing term such that the market and balancing costs are of comparable magnitude.

4.4.1. DEVICE MODEL

We model the group of devices and their aggregate demand as follows, based on the heating model in Appendix B.3.2.5. Let $x(t)$ be the sum of demand at time interval $t \in \mathcal{T}$ (in W), $\mathcal{I}^d = \{1, \dots, 10\}$ the device set (parameter), $x_i(t) \in [0, 1]$ the extent to which heat pump i is turned on in time interval $t \in \mathcal{T}$, $x_i^{\max} = 2000$ is the maximum electricity demand of heat pump i (parameter, in W), and $\eta_i = 5$ the COP of heat pump i (parameter, unitless). Note that all heat pumps in this scenario have the same design parameters (but we keep the indexing by i to be able to change this later on). The variable x_i^+ gives the electricity demand of the auxiliary resistance heating rod of heat pump i at time interval $t \in \mathcal{T}$ (in W), $s_i(t)$ the SoC of the buffer associated with heat pump i after time interval $t \in \mathcal{T}$ (in J), $s_i^{\max} = 8 \text{ kWh} \cdot 1000 \cdot 3600 \text{ (kWh} \rightarrow \text{J)}$ the maximum SoC of the buffer associated with heat pump i (parameter, in J), and $d_i(t)$ the average heat demand that i has to fulfil in time interval $t \in \mathcal{T}$ (parameter, in W). The parameter $\tau = 900 \text{ s}$ gives the length of the time intervals (15 minutes).

The heat demand of i depends on the scenario at hand. However, for notational reasons we omit the indexing by ξ and $\tilde{\xi}$. As we work with a 24 h planning horizon, we can use the heat demand from individual days from the Flex Street dataset to make distinct scenarios (which in principle gives us 365 scenarios, but we have to be careful to use the right days of the year, e.g. with respect to seasonal and weekly variations). To actually have significant heat demand, we choose to work with the first three weeks of the year in winter (early January). For reference, we give the complete heat demand for these weeks for the 10 considered households that we use in the experiments in Figure B.2 (p. 279).

With the above, we get the following electricity demand model:

$$x(t) = \sum_{i \in \mathcal{I}^d} x_i^{\max} x_i(t) + x_i^+(t) \quad \forall t \in \mathcal{T} \quad (4.35)$$

$$0 \leq x_i(t) \leq 1 \quad \forall i \in \mathcal{I}^d, t \in \mathcal{T}. \quad (4.36)$$

Furthermore, the auxiliary resistance heating rod can supply as much heat as necessary:

$$0 \leq x_i^+(t) \quad \forall i \in \mathcal{I}^d, t \in \mathcal{T}. \quad (4.37)$$

The heat pump, auxiliary resistance heating rod, and the demand contribute to the SoC according to:

$$s_i(t) = s_i(t-1) + \tau(\eta_i x_i^{\max} x_i(t) + x_i^+(t) - d_i(t)) \quad \forall i \in \mathcal{I}^d, t \in \mathcal{T} \quad (4.38)$$

$$0 \leq s_i(t) \leq s_i^{\max} \quad \forall i \in \mathcal{I}^d, t \in \mathcal{T}. \quad (4.39)$$

Note that $x_i^+(t)$ is implicitly bounded by $s_i^{\max} / \tau + d_i(t)$.

To give a realistic simulation, we evenly distribute the initial SoCs $s_i(0)$ for all $i \in \mathcal{I}^d$ (parameter, in J) with a fill level between 20% and 80%: $s_1(0) = 0.2 s_1^{\max}$, $s_2(0) = (0.2 + \frac{1}{|\mathcal{I}^d|-1}(0.8 - 0.2))s_2^{\max}$, $\dots$, $s_{|\mathcal{I}^d|}(0) = 0.8 s_{|\mathcal{I}^d|}^{\max}$. Note that we do not start at a common initial SoC of e.g. 50% to avoid synchronization artifacts. We choose not to use random SoC values (e.g. $s_i(0) \sim \mathcal{U}(0, s_i^{\max})$) to avoid changes in initial energy content between runs that may influence the evaluation.

4.4.2. BINARY (ON/OFF) DEVICES

In the model above, the heat pump can be turned on at every power value in the continuous interval $[0, 2000]$. However, many small scale heat pumps do not offer a full range of demand options, but are only able to switch between turning on and off. We can easily extend the model to support these heat pumps by replacing constraint (4.36) with $x_i(t) \in \{0, 1\}$. This gives a mixed integer variant of the problem.

In our device bidding curve implementations, we also allow partial runs (of less than 15 minutes) to completely fill the buffer, and partial runs that exactly satisfy the current demand. Whereas it is in principle possible to model these control options for on/off heat pumps, we choose to omit these control options from the reference MIP model for simplicity.

4.4.3. OBJECTIVE

Finally, to evaluate the outcome of the control approach, we need a cost function. We choose the cost function in such a way that we can obtain a model for which we can compute the optimal solution directly by solving the QP formulation with a solver such as CPLEX (if we would choose a cost function that we are practically interested in, then we may not be able to determine a reference optimal solution). This solution serves as a lower bound on what can be reached in practice, and allows us to effectively evaluate the approach. This is a lower bound on the true minimum cost in the context of our evaluation, because the dispatch mechanism may also have inherent losses (introduce constraints that hide the optimal solution), which may differ between scenarios and bidding strategies. That is, even though we try to evaluate just the supply offer strategy, what we measure is also influenced by the dispatch mechanism, the choice of device bidding functions, and the chosen

scenarios. Although our evaluation only considers a quadratic cost function, the approach should work similarly for most other cost functions. However, we have limited experience with other cost functions in this context, and therefore the approach may perform different from the cases that we evaluated. Note that the lower bound model that we consider here is more detailed than the model that we discuss in Appendix B.6, and can thereby give a tighter bound.

Let $\mathbf{x} = \langle x(1), \dots, x(n_t) \rangle$ be the total electricity demand over time as given by (4.35), $\mathbf{c}$ a corresponding price vector (in €/J), z the cost of the solution, and β a small quadratic balancing term which avoids large exchange peaks. We use 15 minute time intervals ($\tau = 900$ s), take $\mathbf{c}$ from an APX market clearing price dataset, and choose $\beta = 10^{-11}/|\mathcal{I}^d|$ (in $\text{€} \cdot \text{W}^{-2} \cdot \text{s}^{-1}$) as a tradeoff between price and balancing optimization. Based on these parameters, we specify the main part of the objective as $\tau(\mathbf{c}^\top \mathbf{x} + \beta \mathbf{x}^\top \mathbf{x})$. However, to avoid the impact of some border situations, some additional aspects have to be taken into account and are integrated in the objective.

A common problem in optimization over a finite time horizon is that the optimal solutions “sell out” at the end of the horizon. In our case, this means that all buffers are empty at the end of the day. To discourage this behaviour, we set a target SoC for the end of the day. To avoid feasibility problems (it is hard to reach the target SoC *exactly*), we do not introduce hard constraints for the SoC at the end of the day, but use a penalty function z_s to conservatively assess the cost of the mismatch. If the SoC is higher than the target s^* , then we can sell the energy for a low price c_{sell} (in €/J), and if the SoC is lower than the target, then we can buy energy at a high price of c_{buy} (in €/J). We choose $s^* = \sum_{i \in \mathcal{I}^d} s_i(0)$, $c_{\text{buy}} = \max \mathbf{c}$, and $c_{\text{sell}} = \min c / \max_{i \in \mathcal{I}^d} \eta_i$: the buffers should together have the same energy content at the end of the day, any deficit can be bought at the highest price, and surplus may be sold at the lowest price (while accounting for the heat pump COP). In our case, all $s_i^{\max}$ are equal, which gives $s^* = 1/2 |\mathcal{I}^d| s_i^{\max}$. We consider the aggregate SoC at a neighbourhood level and do not specifically manage the balance of SoC between individual buffers, i.e. the cost function does not distinguish between a case with two half-full buffers, and a case with one full buffer and one empty buffer (of equal size).

Based on the above, we now define the objective as a minimization problem with the following cost function:

$$\min z \quad (4.40)$$

$$z = \tau(\mathbf{c}^\top \mathbf{x} + \beta \mathbf{x}^\top \mathbf{x}) + z_s\left(\sum_{i \in \mathcal{I}^d} s_i(n_t)\right) \quad (4.41)$$

$$z_s(s) = \begin{cases} (s - s^*)c_{\text{sell}} & \text{if } s \geq s^* \\ (s^* - s)c_{\text{buy}} & \text{if } s < s^* \end{cases} \quad (4.42)$$

With the model that we presented up to here, we can in principle determine the optimal control for all devices. However, the direct use of this approach has some disadvantages in practice: some internal parameters of the problem, and in particular the heat demand, are in principle not known, but may be estimated from historical data. The heat demand volume determines (bounds on) the electricity demand volume, and thereby limits the possible demand profiles that we can make.

We refer to this as internal uncertainty: we do not exactly know yet what flexibility we have in the future. Also, (4.41) represents “everything” that is outside of our control at a higher level, of which the prices $\mathbf{c}$ are the main uncertain component. We refer to this as external uncertainty: we do not exactly know yet what flexibility others want from us in the future. We can address this uncertainty either by increasing our understanding of the world (which decreases the uncertainty, e.g. by improved forecasting) or by developing uncertainty tolerant control policies (which makes us indifferent to the uncertainty). We focus on the second approach. Also, we focus on the internal uncertainty, and leave the external uncertainty as a topic for future work. As discussed before, these uncertain aspects motivate a dynamic dispatch approach, such as an online auction dispatch, which is hard to address within an optimization problem. This dispatch approach removes most control freedom, which paradoxically makes the optimization problem *harder* (if we choose an MIQP optimization approach), as we need to account for the particular control options that the dispatch mechanism exposes. That is, the optimization problem misrepresents dispatch dynamics.

4.4.4. DETERMINISTIC VARIANT

Although the heat demand is in principle uncertain, we first assume that all parameters, and in particular the heat demand, are completely known. This gives a deterministic variant of the problem, with $\tilde{\Xi} = \Xi$ and $n_{\Xi} = 1$. As we want to have ample heat demand, we pick scenarios from winter. For the deterministic variant, we just choose to use the first day from the dataset (conveniently, the dataset starts at January 1st in winter, and does not have the concept of holidays). A more extensive evaluation should consider more than this one single day, although we have no reason to believe that the results will be much different.

To give an impression of the scenario at hand, we present some values for this scenario. The chosen day has a total heat demand of 319 kWh for 10 households, and thereby a (minimum) electricity demand of 63.8 kWh ($\eta_i = 5$). The average APX electricity price on this day is €0.044/kWh (which is much lower than the usual retail electricity prices). If the demand would be served directly at the time of use at full efficiency (without the use of the auxiliary resistance heating rod), this would cost €4.41 (€2.64 in commodity costs, i.e. $\tau(\mathbf{c}^\top \mathbf{x})$ in (4.41), and €1.77 in balancing costs, $\tau\beta(\mathbf{x}^\top \mathbf{x})$ in (4.41)). The QP solution shows that the cost of the optimal solution is bounded from below to €3.44 for a 24 h horizon with 10 devices.

For the integer variant, it is not feasible to determine an optimal solution within reasonable time with an MIQP, even without considering the dispatch mechanism. The reason for this is that the problem has a large number of binary variables, which makes the problem intractable. Although it may be possible to rewrite the formulation to make it more suitable for an integer solver or to use a DP approach, we have not pursued this approach. Instead, we use the QP solution as an underapproximation of the MIQP solution (the cost of the QP optimal solution is a lower bound for the cost of the MIQP optimal solution).

4.4.5. STOCHASTIC VARIANT

Our main interest lies in the stochastic variant of the problem, where we assume that the heat demand is uncertain. To model this uncertainty, we evaluate n_{Ξ} scenarios (i.e. possible execution realizations) with a specific heat demand vector. Each of these scenarios corresponds to a single day of heat demand in the scenario dataset. We assign equal weight to each of the outcomes from the scenarios. In the planning phase, we need to have a set of scenarios $\tilde{\Xi}$ that is representative of Ξ (i.e. a forecast). We choose $n_{\tilde{\Xi}}$ different days from the dataset to make up $\tilde{\Xi}$. A pragmatic choice for $\tilde{\Xi}$ that can readily be obtained in practice is the history of the $n_{\tilde{\Xi}}$ days that precede the days in Ξ . In our experiments, we choose the first $n_{\tilde{\Xi}}$ days of the dataset as the history for planning, and the next n_{Ξ} days as the possible demand realizations during execution. As the dataset has a strong weekly pattern, we choose two different settings: $n_{\Xi} = n_{\tilde{\Xi}} = 7$ and $n_{\Xi} = n_{\tilde{\Xi}} = 10$ (although we choose $n_{\Xi} = n_{\tilde{\Xi}}$, there are no technical limitations that require these parameters to be equal). For $n_{\Xi} = n_{\tilde{\Xi}} = 7$, this means that we choose the same days of the week for the demand history for the planning as for the possible execution realizations. With $n_{\Xi} = n_{\tilde{\Xi}} = 10$, this is not the case, and thereby we may over- and underrepresent some days of the week (i.e. the realization set Ξ contains two Saturdays and the forecast set $\tilde{\Xi}$ contains one Saturday). Note that we do not make a direct correspondence between the days in the forecast scenario set and the days in the realization set. A real demand forecast may give a better prediction of demand, as the type of day at hand is known and we may incorporate further information (e.g. the weather forecast).

For $n_{\Xi} = 7$, the scenarios in Ξ have an average heat demand of 603 kWh over a 24 h period, and thereby a minimum electricity demand of 121 kWh (with $\eta_i = 5$). The heat demand is substantially higher and highly concentrated in 2 out of 7 scenarios (which correspond to Saturday and Sunday). If the demand would be served directly at the time of use at full efficiency (i.e. the baseline solution), even at times when the heat demand exceeds the maximum output of the heat pump, this would on average cost €8.45 (€4.83 in commodity costs and €3.62 in balancing costs). However, note that this does not correspond to a feasible solution. The QP solution on the test set scenarios gives an average total cost of €8.29. For the QP, we assume that we already know which scenario will occur, and therefore we can make separate decisions for each scenario. The difference between the cost in the QP solution and the baseline solution is remarkably small: we attribute this to the weekend days, where demand is quite high in relation to the buffer size, which makes the scenario infeasible if we do not use the heating rod (which is very expensive and not accounted for in the baseline solution). Also, the heat demand coincides with low electricity prices, which benefits the baseline solution. If we do account for the direct use of the heating rod in the baseline solution (always use the heating rod when the immediate heat demand exceeds the maximum heat production of the heat pump), then we find a cost of €12.84 (€5.90 in commodity costs and €6.94 in balancing costs). This is more in line with the cost difference that we would expect between the QP solution and the baseline solution.

If we now look at the scenarios $\tilde{\Xi}$ that we use for planning with $n_{\tilde{\Xi}} = 7$, these have an average heat demand of 465 kWh, and therefore a minimum electricity demand of 93 kWh. This means that the heat demand, and thereby also the electricity demand, is 30% higher than expected based on the scenarios Ξ . As these amounts differ significantly from the values for Ξ , we have an interesting case to test the robustness of our control approach, i.e. the ability of the control approach to deal with a poorly performing forecasting method.

4.4.6. HEAT PUMP BIDDING FUNCTION

To participate in the dispatch auction, the heat pumps need a bidding function, which dynamically determines when and by how much a device is turned on. In initial experiments, we found that the device bidding strategy can have a significant impact on the objective outcome of the optimization (10 – 15% worse than the QP optimal solution in the deterministic case). We attribute this difference to inefficiency in the dispatch strategy, and in particular the bidding strategy for heat pumps. An important property of the Walrasian auction dispatch mechanism, and of most aggregate dispatch mechanisms in general, is that it ties together the decisions of a large group of devices. Therefore, the proper ordering between devices (which determines who consumes first) is essential to make good use of the available resources. In the case of heat pumps, this means that we should serve heat pumps in the order that their buffers become empty (earliest deadline first), and try to avoid “interference” from devices that do not need to be served yet.

In the following, we present the two bidding strategies that we use in the simulation experiments (Section 4.5): a SoC-based bidding strategy, and a strategy that uses an estimate of the minimum future load (denoted as n/m bidding). Furthermore, to illustrate that defining an appropriate bidding strategy for a device is not always straightforward, we include two failed attempts to define an appropriate bidding strategy: a “naive” SoC-based bidding strategy, and a strategy that estimates the time until we are forced to turn on the heat pump (denoted as l bidding).

If the buffer of the heat pump is near full or near empty, then not all control options are available. Therefore, for each of the presented device bidding strategies, we first determine before the auction at time t the available control range $[\tilde{x}_{i,t}^{\min}, \tilde{x}_{i,t}^{\max}]$ for each device i , which depends on the current SoC $s_i(t-1)$ and the demand $d_i(t)$:

$$\tilde{x}_{i,t}^{\max} = \begin{cases} \min \left(x_i^{\max}, \frac{s_i^{\max} - (s_i(t-1) - \tau d_i(t))}{\tau \eta_i} \right) & \text{if } s_i(t-1) + \tau(\eta_i x_i^{\max} - d_i(t)) \geq 0 \\ x_i^{\max} - \frac{s_i(t-1) + \tau(\eta_i x_i^{\max} - d_i(t))}{\tau} & \text{if } s_i(t-1) + \tau(\eta_i x_i^{\max} - d_i(t)) < 0 \end{cases} \quad (4.43)$$

$$\tilde{x}_{i,t}^{\min} = \begin{cases} 0 & \text{if } s_i(t-1) - \tau d_i(t) \geq 0 \\ -\frac{s_i(t-1) - \tau d_i(t)}{\tau \eta_i} x_i^{\max} & \text{if } s_i(t-1) - \tau d_i(t) < 0 \text{ and } s_i(t-1) + \tau(\eta_i x_i^{\max} - d_i(t)) \geq 0 \\ \tilde{x}_{i,t}^{\max} & \text{otherwise.} \end{cases} \quad (4.44)$$

FIGURE 4.8: Heat pump bidding curve based on current SoC with a static flexible range.

FIGURE 4.9: Heat pump bidding function based on current SoC with a dynamic flexible range. The function switches between “eager” and “reluctant” modes at $p_i^{\text{int}}(t) = 0$ (when the buffer is half-full).

After the auction at time t , we map the auction demand value $x_{i,t}^d(p_t^*)$ back to device decisions $x_i(t)$ (for the heat pump) and $x_i^+(t)$ (for the heating rod) as follows:

$$x_i(t) = \min(1, x_{i,t}^d(p_t^*) / x_i^{\max}) \quad \forall i \in \mathcal{I}^d \quad (4.45)$$

$$x_i^+(t) = \max(0, x_{i,t}^d(p_t^*) - x_i^{\max}) \quad \forall i \in \mathcal{I}^d. \quad (4.46)$$

For the bidding strategies that we present, we also need a variant for the binary version of the problem. As we do not want to come up with a separate bidding strategy for the binary variant, we derive this bidding strategy from the bidding strategy for the continuous variant of the problem. This binary variant takes a step from $\tilde{x}_{i,t}^{\max}$ to $\tilde{x}_{i,t}^{\min}$ at some priority p_t , and we choose the priority p_t such that $x_{i,t}^d(p_t) = (\tilde{x}_{i,t}^{\min} + \tilde{x}_{i,t}^{\max}) / 2$ in the continuous variant of the bidding strategy.

4.4.6.1. SOC BIDDING (STATIC FLEXIBLE RANGE)

Our first “naive” attempt to construct a bidding curve follows a straightforward approach. We set up a decreasing curve from the highest possible to the lowest possible demand value, and use the relative SoC to determine the positioning of this curve. If the SoC is low, then the internal priority is high, and we shift the curve to a high priority level (“to the right”), which motivates consumption. If the SoC is high, then the internal priority is low, and we shift the curve to a low priority value (“to the left”), which avoids consumption, unless the market clears at a low priority level (i.e. plenty of supply is available). Formally, we define the bidding curve as follows:

$$x_{i,t}^d(p) = \text{interp}(\langle 0 - s_i(t-1)/s_i^{\max}, 1 - s_i(t-1)/s_i^{\max} \rangle, \langle \tilde{x}_{i,t}^{\max}, \tilde{x}_{i,t}^{\min} \rangle, p), \quad (4.47)$$

where interp is the function defined in (4.18). We illustrate this bidding function in Figure 4.8, which shows example demand bids of both the “normal” continuous bidding function variant and the derived binary bidding function variant. Note that the binary variant uses only a small part of the available priority range $[-0.5, 0.5]$. This is not a problem as long as all devices use the same range, but does suggest that we should use a dedicated bidding strategy for binary devices in an heterogeneous environment with multiple device types.

Although the bidding strategy is easy to define, early experiments have shown that it performs poorly in practice. A possible explanation for this is that the bidding curve does not “escalate” the problem as the SoC approaches either the upper or lower bound of the buffer (and the latter leads to use of the auxiliary resistance heating rod, which is very expensive in general). Possible workarounds are to use a more narrow flexible range (i.e. a more steep curve, which leads to a smaller active priority range), or to use a larger shifting range (i.e. part of the flexible range falls outside of the window $[-1, 1]$, which leads to must-run or must-not-run behaviour at low or high SoC values, respectively). We did not further investigate these workarounds.

4.4.6.2. SOC BIDDING (DYNAMIC FLEXIBLE RANGE)

In the following, we present a bidding strategy that has a similar base as the previous bidding strategy, which is used in practice and implemented in [119]. This bidding strategy has a dynamic flexible range, where the width of the range depends on the current SoC: the bidding curve has a flexible narrow range near to the upper and lower SoC bounds of the buffer, and a wide flexible range if the SoC is somewhere in the middle. Although the strategy uses the same information as the previous

strategy, it performs much better in our context. We first map the clearing priority p (typical range $[-1, 1]$) to a more convenient range for the bidding function at hand ($[0, 1]$), and denote this “external” priority as p^{ext} . An external priority close to 0 means that the device should run as much as possible, and an external priority of 1 means that the device should try not to run. Similarly, we define an “internal” priority p_i^{int} , which maps the relative SoC onto the range $[-1, 1]$. Although we may also describe the strategy directly in terms of the clearing priority and the buffer SoC, the rescaled versions p^{ext} and p_i^{int} simplify the definition. An internal priority close to -1 means that the device is reluctant to run, because the buffer is nearly full, and an internal priority close to 1 means that device is eager to run, because the buffer is nearly empty. Depending on the internal priority, the device switches between two bidding curve shapes. If $p_i^{\text{int}}(t) \geq 0$ (eager mode), then the heat pump runs at full power, unless p is high. Conversely, if $p_i^{\text{int}}(t) < 0$ (reluctant mode), then the heat pump does not run, unless p is low. At $p_i^{\text{int}}(t) = 0$, these curves have the same shapes. We illustrate this bidding function in Figure 4.9.

More formally, we can describe this bidding function as follows:

$$p^{\text{ext}} = \frac{p + 1}{2} \quad (4.48)$$

$$p_i^{\text{int}}(t) = 1 - 2 \frac{s_i(t-1)}{s_i^{\text{max}}} \quad (4.49)$$

$$y_{i,t}^d(p^{\text{ext}}) = \begin{cases} 1 & \text{if } p^{\text{ext}} \leq 0 \text{ or } p_i^{\text{int}}(t) \geq 1 \\ 0 & \text{if } p^{\text{ext}} \geq 1 \text{ and } p_i^{\text{int}}(t) < 1 \\ 1 & \text{if } p_i^{\text{int}}(t) \geq 0 \text{ and } p^{\text{ext}} < p_i^{\text{int}}(t) \\ 1 - \frac{p^{\text{ext}} - p_i^{\text{int}}(t)}{1 - p_i^{\text{int}}(t)} & \text{if } p_i^{\text{int}}(t) \geq 0 \text{ and } p_i^{\text{int}}(t) < p^{\text{ext}} < 1 \\ 0 & \text{if } p_i^{\text{int}}(t) < 0 \text{ and } p^{\text{ext}} \geq p_i^{\text{int}}(t) \\ 1 - \frac{p^{\text{ext}}}{1 + p_i^{\text{int}}(t)} & \text{if } p_i^{\text{int}}(t) < 0 \text{ and } 0 < p^{\text{ext}} < 1 + p_i^{\text{int}}(t) \end{cases} \quad (4.50)$$

$$x_{i,t}^d(s, p) = y_{i,t}^d(p^{\text{ext}}) \tilde{x}_{i,t}^{\text{max}} + (1 - y_{i,t}^d(p^{\text{ext}})) \tilde{x}_{i,t}^{\text{min}} \quad (4.51)$$

4.4.6.3. I BIDDING

The bidding strategy of Section 4.4.6.2 works well for homogeneous groups of heat pumps with similar demand patterns. However, we believe that relative SoC alone does not accurately indicate priority. For example, if a buffer is two times as large (for the same load), then the relative SoC becomes half as large for the same absolute fill level, which indicates too early that the buffer is near empty. Also, if a household has the same SoC but a higher demand, then the priority should be higher.

A key piece of information in this context is the time until the buffer is forced to charge. This latest charge time depends on the current absolute SoC, the accumulated demand pattern in the near future, and the maximum charge rate. Using a higher charge rate in the current time interval (denoted by the fraction $x_{i,t}$ that the heat pump is turned on) increases the time until the device is forced to charge. We denote this duration until the latest start time as $I(x_{i,t})$, and the expected av-

FIGURE 4.10: Bidding strategy depending on the (expected) future heat demand. The chosen fraction $x_{i,t}$ (with $\tilde{x}_{i,t}^d(p) = x_i \cdot x_i^{\max}$) determines after how long the heat pump must run at full power (i.e. l hours), or must run at full power for n out of m hours. The corresponding priority $p^{\text{bid},l}(x_{i,t})$ is determined by the time duration $l(x_{i,t})$ in the case of l bidding, or $p^{\text{bid},n/m}(x_{i,t})$ by $n(x_{i,t}) / m(x_{i,t})$ in the case of n/m bidding.

erage heat demand of device i in time interval t' as $\tilde{d}_i(t')$, with $t' \in \mathcal{T}^{\text{ahead}} = \{t, \dots, \min(t + n_{i^{\text{ahead}}}, n_t)\}$. Figure 4.10 illustrates this relation. In this figure, we represent the accumulated heat demand by an increase of both the upper and lower SoC bounds of the buffer, which leads to an energy flexibility staircase curve similar to Figure 2.1 in Section 2.3.2.1.4. The buffer is empty when the demand accumulated over time exceeds the amount that we have charged until then (plus the initial SoC). In the example, we choose a charge rate $x_{i,t} = 1/8$. At this charge rate, the buffer will become empty after 2.1 h. However, if the demand may exceed the maximum charge rate, then we should pre-charge to cover for these demand events to avoid the use of the heating rod. We may represent this pre-charging by a correction to the accumulated demand (although we choose to correct it later in (4.54)). The correction depends on the extra energy that can still be charged before the demand event. In a similar way, we may also account for the upper bound of the buffer, but this is left for future work. In the example, if we choose a charge rate $x_{i,t} = 1/8$ and keep using this rate, then we must already start charging at full power after 1.8 h (rather than when the buffer becomes empty) to avoid the use of the heating rod, and keep charging at full power for 0.7 h to meet the demand until $2\frac{1}{2}$ h in the future. This gives the latest charge time $l(1/8) = 1.8$ h, with a charge time $\tilde{n}(1/8) = 0.7$ h and a time $m(1/8) = 2\frac{1}{2}$ h until this demand peak is “over”. If we choose $x_{i,t}$ high enough, then we may never reach the lower SoC bound, leading to $l(x_{i,t}) = \infty$ (in this case, the upper bound restricts the available

options). Note that if the demand (forecast) is constant over time, then the time until the buffer is forced to charge is proportional to the relative SoC, which brings us back to the SoC-based definition of priority, except for a constant that depends on the demand level and the buffer size.

In practice, the future demand is not known in advance. However, since we use only the *accumulated* demand, demand prediction should be feasible with reasonable accuracy. Furthermore, if the demand prediction for a device is inaccurate, then this only causes a misrepresentation of their priority, which usually has a limited effect on the overall dispatching if the number of affected devices is small. In the following, we continue with a bidding function that uses a clairvoyant device-level demand prediction at dispatch time that can look ahead for up to 3 hours (for $\tau = 900$ s, this gives $n_{t^{\text{ahead}}} = 12$).

We define the l bidding strategy as follows. First, $o(t')$ gives the number of time intervals that we have to run the heat pump at full power to satisfy the heat demand until $t' \in \mathcal{T}^{\text{ahead}}$:

$$o(t') = \frac{1}{\eta_i x_i^{\max}} \left(-s_i(t-1) + \sum_{t''=t}^{t'} \tau \tilde{d}_i(t'') \right). \quad (4.52)$$

The value of $o(t')$ is continuous (it does not have to represent whole time intervals), and negative values indicate that the heat pump may be left off until t' . Note that this is essentially a scaled version of the shifted SoC lower bound in Figure 4.10. Next, we choose to charge at a fraction $x_{i,t}$ of full power, and determine the “debt” amount of time $n(x_{i,t}, t')$ that we still need to run at full power to meet demand until t' :⁵

$$n(x_{i,t}, t') = o(t') - x_{i,t} \cdot (t' - t + 1). \quad (4.53)$$

To account for cases where the heat demand exceeds the maximum heat production of the heat pump, we reason backwards to find the “true” value of $o(t')$ and $n(x_{i,t}, t')$, which we denote as $\tilde{o}(t')$ and $\tilde{n}(x_{i,t}, t')$:

$$\tilde{o}(t') = \begin{cases} o(t') & \text{if } t' = \min(n_t, t + n_{t^{\text{ahead}}}) \\ \max(o(t'), \tilde{o}(t' + 1) - 1) & \text{otherwise.} \end{cases} \quad (4.54)$$

The definition of $\tilde{n}(x_{i,t}, t')$ follows by replacing $o(t')$ with $\tilde{o}(t')$ in (4.53). In Figure 4.10, $o(t')$ follows the SoC lower bound, and $\tilde{o}(t')$ follows the corrected lower bound. Now, for a given $x_{i,t}$, we find the first time t' where we are forced to start charging (i.e. $\tilde{n}(x_{i,t}, t')$ becomes positive):

$$l(x_{i,t}) = \begin{cases} \left(\arg \min_{t' \in \mathcal{T}^{\text{ahead}}} \tilde{n}(x_{i,t}, t') > 0 \right) - t & \text{if } \exists t' \mid \tilde{n}(x_{i,t}, t') > 0 \\ \infty & \text{otherwise} \end{cases} \quad (4.55)$$

In Figure 4.10, $l(x_{i,t})$ is given by the time until the projection of $x_{i,t}$ hits the corrected lower bound. Next, we map $l(x_{i,t})$ to an internal priority value:

$$p^{\text{bid}, l}(x_{i,t}) = \text{interp}(\langle 0, c_l \rangle, \langle 1, -1 \rangle, l(x_{i,t})), \quad (4.56)$$

where c_l gives a scaling factor for the latest start time distance, for which we choose $c_l = 3$ h (12 time intervals). This scaling factor gives the minimum latest start time distance until we assign the “very low” priority. For a given value of $x_{i,t}$, we can now determine the corresponding priority value $p^{\text{bid},l}(x_{i,t})$. Next, we use these values to set up a bidding curve: we sample a sequence $p^{\text{bid},l}(x_{i,t})$ for $x_{i,t} \in \left(0, \frac{1}{n_x-1}, \dots, 1\right)$, where n_x is the number of samples, and interpolate the bidding curve:

$$\tilde{x}_{i,t}^d(p) = \text{interp} \left(\left\langle p^{\text{bid},l}(0), p^{\text{bid},l}\left(\frac{1}{n_x-1}\right), \dots, p^{\text{bid},l}(1) \right\rangle, \left\langle 0, \frac{1}{n_x-1}, \dots, 1 \right\rangle, p \right) x_i^{\text{max}}. \quad (4.57)$$

To account for exceptions where the current SoC limits the available range of control options (i.e. the buffer is near full or near empty), we guarantee that the bidding curve respects the limits given by $\tilde{x}_{i,t}^{\min}$ (4.44) and $\tilde{x}_{i,t}^{\max}$ (4.43) by:

$$x_{i,t}^d(p) = \begin{cases} \tilde{x}_{i,t}^d(p) & \text{if } \tilde{x}_{i,t}^{\min} \leq \tilde{x}_{i,t}^d(p) \leq \tilde{x}_{i,t}^{\max} \\ \tilde{x}_{i,t}^{\min} & \text{if } \tilde{x}_{i,t}^d(p) < \tilde{x}_{i,t}^{\min} \\ \tilde{x}_{i,t}^{\max} & \text{if } \tilde{x}_{i,t}^d(p) > \tilde{x}_{i,t}^{\max} \end{cases}. \quad (4.58)$$

Tests show that this bidding strategy does not perform well. In particular, the strategy underestimates “walls” of a sustained high demand, and successive demand peaks. Based on this, we can conclude that we should not only consider the latest start time by which we are forced to charge, but also the amount of work that we are forced to do when we arrive at this time. Although the l bidding strategy is unsuccessful, it forms the basis of a more successful bidding strategy that we present in the following.

4.4.6.4. n/m BIDDING

Next to considering l , i.e. the time until we are forced to charge, we should also account for the amount that needs to be charged. To capture this more generally, we estimate the effect of our control decisions on the load in the near future. To estimate the load, we consider where the “debt” n in full power intervals, divided by the time m by which we have to satisfy the load from n , is highest. With this, we consider not only the latest start time, but also further demand peaks in the future. The time m gives the position of the demand peak in the future that is considered to be the most critical. The fraction n/m gives the proportion of time that we have to run at full power to satisfy the peak at m . We can subsequently reduce this “debt” by charging: a higher charging level $x_{i,t}$ leads to a lower n , and if we are closer to the demand peak, then we should use a higher charging level to satisfy the load in time. We call this the n/m heuristic: if we continue to run at a certain charging level $x_{i,t}$, then we still need to charge for n out of m time intervals

5. This equation is slightly inaccurate as $x_{i,t}$ and the full-charge area overlap, but we can ignore this for load estimation purposes.

at full power to clear the highest demand peak (measured in terms of n/m). Note that Figure 4.10 already represents an example for the n/m structure, with $x_{i,t} = 1/8$. If the proposed charging level $x_{i,t}$ is high enough, then the fraction may become negative, which indicates that we charge more than is necessary on average, which decreases the pressure to consume in the future.

The definition of n , or more precisely $n(x_{i,t}, t')$, comes from (4.53). Note that we no longer have to consider $\tilde{n}(x_{i,t}, t')$ in this context. Next, we define the most critical peak $m(x_{i,t})$, i.e. the time in the future that asks for the highest average demand:

$$m(x_{i,t}) = \arg \max_{t' \in \mathcal{T}^{\text{ahead}}} n(x_{i,t}, t') / (t' - t + 1). \quad (4.59)$$

Next, in principle we map the values n/m for a given charge amount $x_{i,t}$ to a priority $p^{\text{bid}, n/m}(x_{i,t})$ according to:

$$p^{\text{bid}, n/m}(x_{i,t}) = n(x_{i,t}) / m(x_{i,t}). \quad (4.60)$$

Now, we could sample (4.60) for some charge amount $x_{i,t}$ to determine the bidding curve. However, we can show that $m(x_{i,t})$ is actually independent of $x_{i,t}$, and exploit this to simplify the bidding curve. This property is very useful for computational purposes: a “brute force” peak selection method has a time complexity of $\mathcal{O}(n_{\text{ahead}} \cdot n_x)$, where n_{ahead} is the look-ahead window and n_x is the number of considered demand options, which can be reduced to $\mathcal{O}(n_{\text{ahead}} + n_x)$ if the critical peak needs to be found only once. Note that in the context of our search algorithms, we invoke the device bidding curve construction method at least $(n_k \cdot n_{\Xi} + n_{\Xi}) \cdot n_t \cdot |\mathcal{I}^d|$ times. If we look back at (4.53) and (4.59), and charge in the value of $n(x_{i,t}, t')$ (with $t' = m(x_{i,t})$), then we see that we can pull $x_{i,t}$ outwards:

$$m(x_{i,t}) = \arg \max_{t' \in \mathcal{T}^{\text{ahead}}} (o(t') - x_{i,t}(t' - t + 1)) / (t' - t + 1) \quad (4.61)$$

$$= \arg \max_{t' \in \mathcal{T}^{\text{ahead}}} o(t') / (t' - t + 1) - x_{i,t}. \quad (4.62)$$

As $x_{i,t}$ affects each t' equally, it does not influence the outcome of $m(x_{i,t})$:

$$m = m(x_{i,t}) = \arg \max_{t' \in \mathcal{T}^{\text{ahead}}} o(t') / (t' - t + 1). \quad (4.63)$$

That is, $m(x_{i,t})$ is independent of $x_{i,t}$. For $n(x_{i,t})$, we now know that we can consider a fixed $t' (= m)$, which means that $x_{i,t}$ gives a proportional change in value to n/m and therefore to the priority value. Therefore, the bidding function only has to consider the upper and lower possible consumption values ($x_{i,t} = 0$ and $x_{i,t} = 1$, unless we are near to the boundaries). Now we can give the bidding curve as follows:

$$p^{\text{bid}, n/m}(0) = o(m)/m \quad (4.64)$$

$$p^{\text{bid}, n/m}(1) = o(m) - 1/m = p^{\text{bid}, n/m}(0) - 1 \quad (4.65)$$

$$\tilde{x}_{i,t}^d(p) = x_i^{\max} \text{interp}(\langle p^{\text{bid}, n/m}(1), p^{\text{bid}, n/m}(0) \rangle, \langle 1, 0 \rangle, p). \quad (4.66)$$

Again, we use (4.58) to derive the “protected” bidding curve $x_{i,t}^d(p)$ from (4.66).

Although the peak that gives the latest start time is often the most critical peak according to m (that is, $l = m - n$), this is not always the case. Consider a case with $s_i(0) = \epsilon$, $x_{i,t} = 0$, and two demand peaks of $x_i^{\max} \eta_i \tau$ (equivalent to running the heat pump at full power for one interval) at $t = 10\tau$ and at $t = 12\tau$ (the first peak occurs in the not too near future, and the second peak follows closely after the first peak). The first peak determines the value $l = 9\tau$, and gives a possible value $n/m = 1/10$. The second peak gives the value $n/m = 2/12$, which is higher than $1/10$ and therefore the most critical peak. This means that $l = m - n$ does not always hold.

The demand values after the optimization horizon are undefined. Although we could assume that the demand after the horizon is 0 (or $\mathcal{T}^{\text{ahead}} = \{t, \dots, n_t\}$ for the last time intervals), we find that this motivates inappropriate behaviour (i.e. empty buffers in the end). Instead, we extend the demand vector with a demand peak of $\frac{1}{2}s_i^{\max}$, which motivates that the buffer is half-full in the end, as is requested by the use case at hand. This should also help for shorter (but not too short) look-ahead windows, as we now try to keep the SoC at a “reasonable” value even if we do not foresee any demand.

4.5. EXPERIMENTS

To evaluate the proposed control approach, we perform simulations with the example use case of Section 4.4. These simulations should give an idea of the performance of the proposed method. Note that both the control approach and the use case have a large number of parameters. Our goal is to find suitable parameters for the control approach that are robust to (or have a straightforward dependency on) changes of the parameters of the use case.

In the following, we first give an overview of the parameter space, and the default parameters that we use for both the search algorithm and the use case (Section 4.5.1). Next, we evaluate the influence of these parameters, and the robustness to changes of these parameters (Section 4.5.2). After this, we conclude our discussion on dispatch centric optimization in Section 4.6.

4.5.1. SEARCH PARAMETERS

To make effective use of the metaheuristic optimization, we need to tune the parameters of the search process to fit to the given problem. Even though it is in principle possible to vary these parameters during the optimization, we assume that all parameters (except for $\hat{\mathbf{x}}$) are determined off-line. Next to this, the use case has a large number of parameters as well. While we can in principle perform a grid search on all possible assignments of parameters, this is highly impractical due to combinatorial explosion of the number of experiments, while individual experiments already have a quite high computational cost (30 – 90 s per run with $n_k = 300$ on a desktop PC with an Intel Core i5-6600 (4 cores, 16 GB of RAM), depending on the experiment at hand). Therefore, for many of these parameters, we have determined default values by experience. Most of these values have already

been given at the point of introduction (e.g. the examples in relation to the supply offer strategy description). We list the parameters, together with a rationale for choosing these parameters, in Appendix C.2.1 (p. 312).

In the experiments, we vary the following parameters. From the use case, we vary between the continuous and binary variant, the prices (c_a and c_b), and whether the planning knows the heat demand data or has to rely on a forecast of this data, i.e. a deterministic variant and stochastic variant. In the stochastic variant, we vary the size of the scenario sets ($n_{\Xi} = n_{\tilde{\Xi}} = 7$ and 10). From the control approach, we vary the optimization algorithm for the economic agent (random optimization, simulated annealing, and genetic algorithm) and the heat pump bidding strategy (SoC and n/m). Note that the demand bidding strategy of the devices is defined independent from the economic agent's supply offer strategy, yet can have a significant impact on the strategy that the economic agent should choose.

4.5.2. USE CASE RESULTS

In the following, we present the results of simulations with the use case of Section 4.4. We first consider the deterministic variant of the use case in Section 4.5.2.1, and continue with the stochastic variant in Section 4.5.2.2 to evaluate the effectiveness of the control approach in a context with uncertainty. After this, we further evaluate the control approach based on these results in Section 4.5.3.

4.5.2.1. DETERMINISTIC VARIANT

We first consider the use case in a deterministic context, i.e. with $n_{\Xi} = n_{\tilde{\Xi}} = 1$ and $\Xi = \tilde{\Xi}$, following Section 4.4.4 and the configuration of the control approach as described in Section 4.5.1.

4.5.2.1.1. Example experiment To illustrate the experiment results, we first present the result of a single experiment run in Figure 4.11 (parameters: continuous variant, n/m bidding, prices c_a , random optimization). The middle plot represents the assignment of $\hat{\mathbf{x}}^{\pi}$ (as in Figure 4.5a), together with the exchange profile $\mathbf{x}^{d,\xi}(\hat{\mathbf{x}}^{\pi})$ that is realized in execution by the control approach (for the single scenario $\xi \in \Xi$), and the optimal exchange profile as determined by the QP. What we see is that the search finds very exaggerated values of $\hat{\mathbf{x}}^{\pi}$, which are an order of magnitude larger than the typical exchange values. Therefore, the bottom plot zooms in on the exchange profiles. In this case, there is a clear correspondence between the optimal exchange profile (from the QP) and the exchange profile that is realized by our control approach, which clearly suggests that we have found a good assignment of $\hat{\mathbf{x}}^{\pi}$. The exchange profile also agrees with the price profile c_a in the upper plot (low prices give high demand), though the small buffer size forces us to consume electricity in a high price period at the end of the day.

There are three reasons for the large values of $\hat{\mathbf{x}}^{\pi}$ in relation to the realized exchange values. First, we have to consider the appropriate part of the priority range that is given by the demand curve. For example, if the desired exchange quantity 10 kW occurs in the demand curve for a priority level 0.1 ($x_t^d(0.1) = 10$ kW), and

FIGURE 4.11: Exchange profile and $\hat{\mathbf{x}}^\pi$ for deterministic variant (middle plot). To make the exchange profile more visible, the bottom plot zooms in on the middle plot. The upper plot gives the corresponding electricity prices.

FIGURE 4.12: Convergence of random optimization for deterministic variant, 5 runs on the same case.

our curve specifies the supply offer values for the priority levels -1 , 0 , and 1 , then we have to choose the supply offer values for $p = 0$ and $p = 1$ in such a way that the desired quantity occurs at $1/10^{\text{th}}$ between the specified points (e.g. $x_{is,t}^s(0) = 2 \text{ kW}$

and $x_{i^s,t}^s(1) = 82 \text{ kW}$ gives $x_{i^s,t}^s(0.1) = 10 \text{ kW}$). This suggests that it may be better to control the priority levels instead of the supply offer quantities in the model of Section 4.2.3 (we further discuss this idea in Section 4.6.1.1.1). Second, supporting this suggestion, the search is often indifferent to some of the selected values, because the relevant flexible range is in a different part of the priority range. In the example, we are indifferent to the value of $x_{i^s,t}^s(-1)$ (although it can not be larger than $x_{i^s,t}^s(0)$). This applies in particular if we consider only a single scenario, but also if the scenario set or the objective is biased (during simulation, we may only see demand curves with very high or very low relevant priority values). This bias is generally appropriate: buffers tend to be full after a low price period, which leads to a low priority. In the earlier example, the supply offer for the priority value -1 has no effect. What we see in these cases (e.g. between 6:00–12:00 at the high priority level and 18:00–21:00 at the low priority level) is that the randomness in the search tends to moderate itself: we randomly take steps both in increasing and decreasing direction, and as the priority values at hand are not relevant, this on average gives no change but may occasionally still give a large value. A third reason (not demonstrated in the figure) is a “must-run” supply offer: if we put in a high supply offer value for every priority value, then we consume as much as possible. This is an appropriate strategy in the case of pure price-based optimization (i.e. buy as much as possible when electricity is cheap). We have also seen instances where large price changes motivate the search to “split” a time period with a steep transition from a very high to a very low supply offer, but this mostly occurs for smaller n_{i^s} .

4.5.2.1.2. Convergence In Figure 4.12, we consider the convergence of the objective value during the execution of the search algorithm (parameters: continuous variant, prices $\mathbf{c}_a$, random optimization). We separately consider the SoC bidding strategy (Section 4.4.6.4) and the n/m bidding strategy (Section 4.4.6.2). We run the search algorithm 5 times on the exact same case. Both cases have the same (lower bound on the) optimal value z^* as determined by the QP formulation (Section 4.4.3). We assess the performance (in terms of cost, as given by the use case) of our control approach relative to z^* , and denote this e.g. as $+30\%$ if $z(\hat{\mathbf{x}}^\pi) = 1.30 z^*$. The thin lines give the cost for the candidate solutions $\hat{\mathbf{x}}^{(k)}$ in each run, and the bold lines at the bottom clamp to the best-known solution $\hat{\mathbf{x}}_{\text{sel}}^{(k)}$ in each run.

We first consider the result for the SoC bidding strategy (Figure 4.12a). In each of the runs, the lowest-seen cost declines rapidly until about iteration $k = 50$ (to $z^* + 10\%$, starting from $z^* + 40\%$ for $\hat{\mathbf{x}}^{(0)}$), and continues to decrease gradually after this (until about $+5\%$ for $k = 300$). We think that this is a good outcome, although a faster convergence would be desirable from a practical perspective. Note that these results do not yet tell anything about the performance of our approach in an uncertain environment, as here we have only attempted multiple times to find a suitable supply offer strategy for a single scenario. The mutations to the supply offer strategy $\hat{\mathbf{x}}^{(k)}$ often have a small impact on the cost outcome. While one could argue that this means that the supply offer strategy is robust to a poor choice of $\hat{\mathbf{x}}^\pi$, we believe that this “robustness” comes from having a lot of (near-)indifferent variables in $\hat{\mathbf{x}}^{(k)}$.

bin.	bid.	prices $\mathbf{c}_a$						prices $\mathbf{c}_b$					
		random		sim. ann.		genetic		random		sim. ann.		genetic	
		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
n	SoC	4.2±1.6%		5.6±2.2%		5.5±1.9%		5.2±1.9%		7.0±2.2%		7.7±2.0%	
n	n/m	2.6±0.8%		6.6±4.6%		3.3±2.0%		2.7±1.6%		4.2±2.6%		2.6±0.6%	
y	SoC	7.7±1.6%		10.2±2.9%		7.5±1.4%		9.3±2.6%		11.4±3.5%		8.4±1.8%	
y	n/m	4.1±1.2%		11.6±9.2%		4.0±0.8%		4.4±1.8%		9.0±4.5%		4.3±0.7%	

TABLE 4.1: Results for deterministic variant, cost penalty in % relative to z^* (mean value μ and standard deviation σ in percentage points of 30 runs).

For the n/m bidding strategy (Figure 4.12b), using the same supply offer optimization procedure, we find similar results. Still, there are notable differences. In comparison to the SoC bidding strategy, the search behaves much more unpredictable, reaching the +10% cost level between $k = 5$ and $k = 50$. The subsequent improvement after this behaves unpredictable as well. Nevertheless, the found solutions at $k = 300$ on average have a lower cost ($z^* + 3\%$) than those of the SoC bidding strategy. We also see that mutations often give a large change to the objective cost, which means that the variables of $\mathbf{x}^{(k)}$ are indeed used. However, the slow convergence shows that the search may have problems to select the right combination of variables to change. Note that the example in Figure 4.11 gives the exchange profile outcome for the selected solution at $k = 300$ from this case in one run. Here, we give only the cost outcome, but also show the rejected candidate solutions.

In general, the search converges well in 100 – 150 iterations. However, exceptions where the search fails to converge, or only converges very slowly, sometimes occur. A straightforward solution to this problem is to run the procedure multiple times, and then pick the best outcome. The cost of this solution is extra computation and communication cost, which is linear in the number of attempts (although we can perform these attempts in parallel). Note that we can in principle decide dynamically whether we should restart or continue the procedure. If we can determine $\tilde{z}^*$ at run-time (e.g. with a QP), then we may even set an (estimated) cost performance cut-off.

4.5.2.1.3. Results In Table 4.1, we summarize the simulation results for the deterministic variant of the problem, with the parameters as discussed in Section 4.5.1. The field “bin.” indicates if we consider the continuous variant (n) or the binary variant (y) of the heat pump in the use case. The field “bid.” indicates which bidding strategy we use for the heat pumps (SoC-based as in Section 4.4.6.2, or n/m , as in Section 4.4.6.4). We furthermore split out the results according to the metaheuristic search algorithm that was used to find $\mathbf{x}^*$ (random optimization, simulated annealing, or genetic algorithm). For validation, we use two different price vectors $\mathbf{c}_a$ and $\mathbf{c}_b$. As we discussed before shortly, the outcomes between experiments may (and do) differ measurably between experiment runs, even with the exact same input parameters.

Therefore, to gain confidence that occasional mediocre results (or their absence) do not bias our evaluation, we run all experiments 30 times, which should be more than enough. The table presents the mean value and standard deviation of these experiments, as the extra cost of $z(\tilde{\mathbf{x}}^\pi)$ relative to the corresponding value of z^* . Note that the corresponding value of z^* here only depends on the pricing scheme, as we do not have a separate solution for the binary variant and the bidding strategy of the devices is not considered in z^* .

First, we focus on the continuous variant with prices $\mathbf{c}_a$ (the upper left quadrant of Table 4.1). We start with the results for the random optimization algorithm, which were also considered in Section 4.5.2.1.2 (and in Figure 4.12). Here, we again see that our supply offer strategy gives results that are competitive with the optimal solution, and that the n/m device bidding strategy outperforms the SoC bidding strategy. What is interesting to see is that whereas the spread of the cost of the intermediate (discarded) solutions is higher for the n/m bidding strategy than for the SoC bidding strategy, the spread of solution cost for the end result with the SoC bidding strategy is higher than for the n/m bidding strategy.

Next, we consider the results for the simulated annealing algorithm, and use the random optimization algorithm as a baseline. We see that it gives a higher mean value, and a higher standard deviation as well. This shows that for the problem at hand, randomly choosing non-improving solutions rarely pays off, or the iteration count is too low to compensate for this “wandering” behaviour. In the n/m case, poor results occur particularly often as candidates with a high cost are common (as seen in Section 4.5.2.1.2). In a further investigation, we found that the algorithm performs best on this problem for $c_{\text{accept}} \rightarrow \infty$, i.e. if we do the same as with random optimization.

The genetic algorithm gives mixed results: in some cases, the genetic algorithm performs slightly better than random optimization, and slightly worse in other cases. One of the disadvantages of this approach is that it may keep using “old” solutions for too long, which hampers progress. However, an advantage is that a single “mis-step” gives less risk to become trapped in a local minimum, as we can still explore other promising directions. We think that it may be promising to further explore the genetic algorithm approach, and in particular try to find effective genetic combination operators.

For the binary variant of the problem (bottom left quadrant), the cost performance is worse than for continuous variant. This is expected, because we can now control the group of heat pumps only in increments of 2 kW, and the reference z^* which uses a relaxed version of the problem does not consider this restriction. Considering this, we think that the control gives reasonable results. Again, simulated annealing performs worse than random optimization. In this case, the genetic algorithm performs slightly better than random optimization. We think that the risk of local minima is slightly higher in this case in comparison to the continuous variant, because the devices are controlled in large increments. This difference may go away if we consider a larger device group, which makes the consumption of an individual device relatively insignificant.

To validate the results, we also consider a different price vector $\mathbf{c}_b$ (right half of Table 4.1). We find comparable results, although the SoC bidding strategy appears to perform slightly worse than for $\mathbf{c}_a$. The similarity of the results gives some confidence that our approach works as intended.

4.5.2.2. STOCHASTIC VARIANT

In the preceding, we have demonstrated that our control approach produces competitive results in a deterministic environment. We now evaluate the control approach for the stochastic variant of the use case, i.e. with $n_{\Xi} = n_{\tilde{\Xi}} \in \{7, 10\}$ and $\Xi \neq \tilde{\Xi}$, following Section 4.4.5. The discussion follows the same structure as for the deterministic variant. We first consider examples for $n_{\Xi} = 7$ in Section 4.5.2.2.1 and Section 4.5.2.2.2.

4.5.2.2.1. Example experiment We first present the results of a single run with $n_{\Xi} = 7$ in Figure 4.13. In this case, a single assignment of $\hat{\mathbf{x}}^{\pi}$ leads to 7 different possible realizations of the exchange profile (i.e. scenarios). The figure shows 3 of these realizations, alongside the QP optimal exchange profile for each of these scenarios. We find that the exchange profiles again follow the optimal exchange profiles quite well, although the differences are larger than in the deterministic variant. Also, the exchange profiles follow the price curve less distinctly. The difference is larger, because the supply offer strategy now has to accommodate multiple scenarios (from $\tilde{\Xi}$), which makes “micro-management” difficult unless we choose a high number of priority levels n_{p^*} in the supply offer curve. Furthermore, the exchange profiles in Figure 4.13 result from control in the scenarios Ξ , which are not known in the planning phase and may differ significantly from $\tilde{\Xi}$ (as discussed in Section 4.4.5). This demonstrates the robustness of our control approach.

If we look at the values of the supply offer assignment $\hat{\mathbf{x}}^{\pi}$, then we find that these tend to be in a more reasonable range than for the deterministic variant. The reason for this is that there are less “indifferent” values (as we have multiple applicable clearing priorities), and that we have to fit the supply offer curves to multiple demand curves (for each scenario in $\tilde{\Xi}$). We also see that “indifferent” values in $\hat{\mathbf{x}}^{\pi}$ still exist, i.e. values in $\hat{\mathbf{x}}^{\pi}$ that do not have an effect in $\tilde{\Xi}$ may be assigned arbitrary values (see Section 4.5.2.1.1), and that these may indeed give adverse control results if these values are active in Ξ . We think that the proposal to control the priority levels instead of the supply quantity (Section 4.5.2.1.1) may help to improve robustness in these cases. Alternatively, we could use a more tapered supply offer curve for unexplored priority levels.

4.5.2.2.2. Convergence In Figure 4.14, we consider the convergence behaviour in the stochastic variant of the problem (parameters: continuous variant, prices $\mathbf{c}_a$, random optimization, $n_{\Xi} = n_{\tilde{\Xi}} = 7$). We run the search algorithm 5 times on the exact same case. Each curve corresponds to one of these runs, and gives the weighted cost for 7 scenarios.

FIGURE 4.13: Exchange profile for stochastic variant with $n_{\Xi} = 7$ and prices $\mathbf{c}_a$. The lower plot shows 3 out of 7 exchange profiles for a single run (bold lines), their corresponding QP optimal exchange profile (thin, dashed lines), and the electricity prices (upper plot). Note that most values of the supply offer assignment $\mathbf{x}^{\pi}$ (which determine $x_{i^s,t}^{s,\pi}(p)$) are outside of the displayed range.

FIGURE 4.14: Convergence of random optimization for stochastic variant, 5 runs on the same case.

First, we consider the lower bundle of cost curves in both Figure 4.14a and Figure 4.14b. These curves give the value of $\tilde{z}(\mathbf{x}^{(k)})$ during the optimization procedure (one curve per run), i.e. the resulting cost for the scenarios $\tilde{\Xi}$ which are used for planning. Note that, in contrast to Figure 4.12, we have omitted the bold curves for the best-known solution $\tilde{z}(\mathbf{x}_{\text{sel}}^{(k)})$. We show the current expected cost $\tilde{z}(\mathbf{x}^{(k)})$ of the search together with the expected cost $\tilde{z}^*$ for the QP optimal assignment for $\tilde{\Xi}$. If we compare these plots to Figure 4.12, then we see comparable results, although the results here are less pronounced. The reason for this is that the cost is averaged over multiple scenarios. Also, the convergence appears to be slightly slower, because the search now has to accommodate multiple scenarios. The average cost level is higher here than for the deterministic case (about €2/day), because the average heat demand is higher. Note that the lower curves in principle still describe a quasi-deterministic environment (we know all possible outcomes, although we do not know which one will occur), but we now have a single control strategy for multiple scenarios $\tilde{\Xi}$.

In the following (and in Section 4.5.2.2.3), we evaluate how the control strategy that we optimized for $\tilde{\Xi}$ “holds up” if we apply it to Ξ . In simulation, we determined the expected real cost $z(\mathbf{x}^{(k)})$ for intermediate $\mathbf{x}^{(k)}$ during the search procedure. In both Figure 4.14a and Figure 4.14b, the resulting values of $z(\mathbf{x}_{\text{sel}}^{(k)})$ during the optimization procedure are given together with the value of z^* . Even though the values of $z(\mathbf{x}^{(k)})$ are significantly higher than $\tilde{z}(\mathbf{x}^{(k)})$ (about €3/day, due to the higher heat demand), the trend is that z decreases along with $\tilde{z}$, which is what we intended. However, the decrease of z is certainly not monotonic, as we may find assignments that are better for $\tilde{z}$ but worse for z , and we have no way to observe when this happens. This particularly applies to the SoC bidding strategy, and less for the n/m bidding strategy. A possible explanation for this is that the (current implementation of the) n/m bidding strategy “cheats” by exploiting coarse knowledge about the scenario $\xi \in \Xi$ at hand. This may also explain why the cost is also already much lower with the n/m bidding strategy before the actual search starts (for $z(\mathbf{x}^{(0)})$). The value of z tends to more or less stabilize after $k = 150$ for the case at hand.

For the selected assignment $\mathbf{x}^{\pi}$ in each case, the *relative* extra cost is about twice as high as for the deterministic case: 8% for the SoC bidding strategy and 4% for the n/m bidding strategy. We think that these are good performance figures, given the poor quality of the used forecast.

4.5.2.2.3. Results In Table 4.2, we summarize the simulation results for the stochastic variant of the problem, with parameters as discussed in Section 4.5.1. The headers follow the structure that we described in Section 4.5.2.1.3, and we again perform 30 runs for each case. To validate the results, we consider scenario sets with $n_{\Xi} = n_{\tilde{\Xi}} = 7$ and $n_{\Xi} = n_{\tilde{\Xi}} = 10$. In the table, the column n_{Ξ} indicates which of these scenario sets Ξ and $\tilde{\Xi}$ is used (with $n_{\tilde{\Xi}} = n_{\Xi}$).

We first focus on the continuous variant with prices $\mathbf{c}_a$ with $n_{\Xi} = 7$ (upper left quadrant, odd rows in Table 4.2). We start with the results for the random optimization algorithm, which corresponds with Section 4.5.2.2.2. As discussed

bin.	bid.	n_{Ξ}	prices $\mathbf{c}_a$						prices $\mathbf{c}_b$					
			random		sim. ann.		genetic		random		sim. ann.		genetic	
			μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
n	SoC	7	8.4±1.2%		9.3±2.1%		8.9±1.1%		10.3±0.9%		11.0±1.8%		10.6±1.1%	
n	SoC	10	7.6±1.1%		8.5±1.5%		8.2±1.1%		10.3±1.4%		11.0±1.8%		10.4±1.6%	
n	n/m	7	4.1±0.5%		4.3±1.2%		5.0±0.7%		6.3±1.5%		6.1±2.3%		6.9±1.4%	
n	n/m	10	4.2±0.5%		4.5±1.0%		4.7±0.8%		5.4±1.3%		5.2±1.2%		5.9±1.2%	
y	SoC	7	11.4±2.4%		12.8±3.4%		10.8±1.9%		13.5±2.5%		14.2±2.4%		13.0±1.4%	
y	SoC	10	10.6±2.5%		10.7±2.7%		11.0±2.0%		12.6±1.6%		13.6±2.5%		12.2±1.5%	
y	n/m	7	5.1±1.2%		7.4±5.5%		4.6±0.7%		6.7±2.1%		7.2±3.4%		6.1±1.7%	
y	n/m	10	5.7±1.3%		6.2±2.5%		5.2±1.0%		6.8±2.1%		8.9±6.0%		6.4±1.8%	

TABLE 4.2: Results for stochastic variant, cost penalty in % relative to z^* (mean value μ and standard deviation σ in percentage points of 30 runs).

there, the relative cost penalty is about twice as high as for the corresponding deterministic problem with $n_{\Xi} = 1$ in Table 4.1, which may be considered as a good result, given the poor quality of the forecast. Although the variation of the resulting costs is slightly lower than for the deterministic problem, we think that this should be attributed to the averaging over the scenarios. The simulated annealing algorithm again performs worse in general than the random optimization algorithm, but the difference is smaller than for the deterministic variant of the problem that we evaluated in Section 4.5.2.1.3. If we look back to Figure 4.14, then we see that the cost peaks of poor assignments are less pronounced than in the deterministic variant, which means that the severity of randomly accepting a worse assignment is lower. For the genetic algorithm, the results are in line with the results for the deterministic variant. If we now look at the results for the binary variant (lower left quadrant, odd rows), the results are in line with the deterministic variant as well. In particular the n/m bidding strategy seems hardly affected, which may be attributed to the extra information that this bidding strategy uses, as explained in Section 4.5.2.2.2.

If we consider the same case for a different price vector $\mathbf{c}_b$ (right half of Table 4.2, and still considering the odd rows $n_{\Xi} = 7$), then we see that the control approach performs less well than for the price vector $\mathbf{c}_a$ that we considered earlier: the costs relative to the corresponding z^* are consistently higher by about 2 percentage points. To explain where this cost increase comes from, in Figure 4.15 we present the exchange profiles for the same case and the same scenarios as Figure 4.13, but now optimized for the prices $\mathbf{c}_b$. First, we notice that the average price level is much higher than for $\mathbf{c}_a$. This means that the relative weight of the balancing component becomes less, and thereby leads to steeper transitions between exchange levels in the optimal solution. In the exchange profile plot, we see that these transitions indeed give losses with our supply offer strategy. To follow these transitions, we could introduce enough time points n_{t^s} to obtain a high temporal resolution, or introduce

FIGURE 4.15: Exchange profile for stochastic variant with $n_{\Xi} = 7$ and prices c_b , for the same 3 scenarios as in Figure 4.13.

mutation operators that operate on wider “chunks” of time intervals in the original time interval domain ($n_{t^s} = n_t$), although these both come at the expense of higher search effort. Also, we could adapt the choice of time points to the structure in the price vector. Instead, we propose to make the supply offer curve dependent on the current price, and we further discuss this idea in Section 4.6.1.3.1. In the plot, we see that our control strategy ramps down slowly between 8:00–10:00, which leads to consumption during the price peak, and we fail to ramp up immediately at 12:00 when prices are low again. Also, we see some peaks resulting from the forced use of the auxiliary resistance heating rod that could have been prevented (at 13:00 and 19:00). Looking back at Table 4.1, we do not see the same trend for the deterministic variant that we considered. We think that this has to do with the difference in heat demand between these scenarios, which is much lower for the deterministic variant, and better allows to “pre-charge” well in advance of the price peak.

Finally, we look at the results for $n_{\Xi} = 10$ (even rows in Table 4.2), and compare these to the results for $n_{\Xi} = 7$ (odd rows). We see that this increase in the number of considered scenarios seems to increase the solution quality slightly (0.5–1 percentage points lower cost), though this does not occur consistently. Note that the forecast error is comparable to $n_{\Xi} = 7$ (33.5% higher heat demand than expected). The consistency of the results between $n_{\Xi} = 7$ and $n_{\Xi} = 10$ demonstrates the robustness of the control approach that we have developed.

bin.	bid.	n_{Ξ}	prices c_a						prices c_b					
			random		sim. ann.		genetic		random		sim. ann.		genetic	
			μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
n	SoC	7	7.8±0.8%		8.2±1.5%		8.6±1.3%		11.1±0.4%		11.5±0.5%		11.5±0.7%	
n	SoC	10	7.4±0.6%		7.8±0.9%		8.6±1.1%		10.8±0.3%		10.9±0.5%		11.0±0.4%	
n	n/m	7	4.0±0.7%		3.8±0.5%		4.9±0.9%		4.7±0.8%		5.2±1.7%		6.0±1.2%	
n	n/m	10	4.4±0.8%		5.5±2.0%		5.6±1.3%		5.3±1.3%		5.0±1.0%		5.9±1.1%	
y	SoC	7	9.6±1.3%		10.7±2.2%		9.6±1.4%		12.4±1.3%		13.8±2.0%		11.9±0.9%	
y	SoC	10	9.8±1.4%		10.3±1.8%		9.8±1.3%		11.9±1.0%		12.4±1.2%		11.8±0.5%	
y	n/m	7	4.6±0.8%		6.4±1.9%		5.1±0.7%		5.0±1.2%		7.8±3.6%		5.2±1.1%	
y	n/m	10	5.7±1.3%		6.3±2.8%		4.9±0.9%		5.2±0.8%		7.0±3.0%		4.9±1.1%	

TABLE 4.3: Results for stochastic variant with swapped scenarios, cost penalty in % relative to z^* (mean value μ and standard deviation σ in percentage points of 30 runs).

4.5.2.2.4. *Results (swapped scenarios)* In the case of Section 4.5.2.2.3, we consider the cost in a scenario set where the planning underestimates the heat demand. To check whether the approach also works well in the case of an overestimation of demand, we should consider a different scenario set. By swapping the contents of Ξ and $\tilde{\Xi}$, we can straightforwardly derive such a case with an overestimation of heat demand. In Table 4.3, we give the simulation results for this case. We find that the results are remarkably similar to the results in Table 4.2, which leads us to the conclusion that our control approach is robust in both cases of underestimation and overestimation of demand.

4.5.3. EVALUATION

The simulations in this section demonstrate that the developed control approach is robust, and still gives reasonable control results (i.e. low costs) when the forecast errors in the planning phase are large. The control result is competitive with the QP optimal solution based on full information of the future and the scenario at hand. We think that the robustness may be further improved if we choose to shift the priority levels of the supply offer strategy instead of the offer quantities (see Section 4.6.1.1.1). The control approach in principle uses very limited information on the problem at hand, and sets up a supply offer strategy with a coarse structure. This coarse structure helps to prevent overfitting of the strategy to the planning scenarios, but also limits the types of control strategies that we may describe, and thereby restricts the objective performance on some cost functions (e.g. with rapid price changes). We think that we can remove or reduce this restriction by considering dynamic information in the offer strategy (e.g. the current price), at the expense of increased complexity.

The metaheuristic search algorithms manage to find a suitable assignment for the parameters of the supply offer strategy, with very little information on the problem at hand. However, the computational cost of the search is quite high, which means that the size of the problems that we can address in a practical setting with the current approach is limited. Nevertheless, day-ahead planning on a neighbourhood level with hundreds of households is still feasible. We performed the experiment presented in the previous sections also for an instance with $|\mathcal{I}^d| = 100$, and achieved comparable cost results (+ 4.6% in comparison to QP for $n_{\Xi} = 7$, n/m bidding, random optimization, continuous variant). While the search algorithms are in principle easy to develop, finding an appropriate configuration and operators for the search algorithms takes considerable “trial and error” effort within the simulations. We tend to find the best results with the more simple search strategies.

The current approach may also incur high communication costs in a decentralized implementation. Regarding the choice of search algorithms, more attention could be paid to their specific properties in the light of the problem at hand, as opposed to choosing the most well-known algorithms. To make the control approach efficient, we think that the iteration count should be much reduced, or the effort for one iteration should be much reduced. The key to achieve this is to make use of the structure in the problem. Although well chosen, more advanced metaheuristic search algorithms may help to reduce the search effort and to find a less “crude” supply offer strategy, we expect that it is worthwhile to develop a dedicated heuristic search algorithm that exploits domain knowledge of this specific application.

A major point that has come to our attention during the development of the control approach is the importance of using good device bidding strategies. As the simulation results clearly indicate, the bidding strategy can have a significant impact on the control result that we achieve. Looking at the problem from a mathematical optimization perspective, we think that the bidding strategy adds constraints that may make the optimal solution for the original problem (without a dispatch mechanism) unavailable. The priority ordering of the control options in the aggregate bidding curve of the devices determines the possible combinations of control options that we can choose from (i.e., it may not be possible to turn on a device unless some other devices are turned on as well). Therefore, the bidding functions should be chosen such that we obtain suitable combinations of control options. We furthermore think that the bidding strategies of some devices may benefit from a local load forecast (i.e. may lead to a more realistic representation of urgency), as demonstrated by the n/m bidding function that we proposed. A topic that has been left untouched here is the control of a cluster with heterogeneous devices. This gives extra challenges, as the bidding strategies of these devices have to be made compatible with each other. Alternatively, we could in principle consider the device bidding strategies as variables in the search procedure, but we expect this to be prohibitively expensive. A more structured approach may consider the co-design of the demand bidding strategies with the supply offer strategy. By this, we do not take the bidding strategies as given, but choose the bidding function such that it fits with the control result that we want to achieve. This requires control over the bidding strategy of devices. We propose an example that uses this approach in Section 4.6.1.4.1.

Reflecting on the experiments themselves, it may be worthwhile to consider scenarios with more diverse devices (also with different device types or device configuration), having different cost function configurations (price vectors), and using different search algorithm configurations. However, this is expensive in terms of simulation run-time. We think that a more extensive evaluation is asked for after the known design issues are addressed: the efficiency of the search algorithm should be improved, the approach should allow for more dynamic behaviour (e.g. adapt the supply offer strategy to the electricity price), and we should have a more structured approach to set up good bidding strategies for the devices. Also, the simulation results suggest that it may be better to control the priority levels instead of the supply quantities in the supply offer strategy model. Nevertheless, the simulation results give confidence in the idea of the presented control approach.

4.6. CONCLUSION

In this chapter, we have presented a control approach where we take a dynamic dispatch mechanism as a starting point, and realize the planning by influencing this dispatch mechanism. Hereby, our control approach enables the planning of devices that are controlled through a Walrasian auction dispatch mechanism. With this approach, a DSM aggregator can optimize the electricity exchange with the grid of these devices over time, according to some given objective. The aggregator manages the exchange by choosing the shape of the supply offer curve that it presents to the auction mechanism. The implementation of the optimization approach relies on a simulation model of the bidding behaviour of the devices, and uses a metaheuristic algorithm to optimize the supply offer strategy. To reduce the search effort for the metaheuristic algorithm, we use a simplified supply offer strategy model. This model is chosen such that it is robust to changes in demand of the devices.

The control performance of the approach has been evaluated with a use case in Section 4.4, which is derived from Flex Street (Appendix B). As a reference, we determine the optimal solution without considering the dispatch mechanism. The simulations show that our control approach is competitive with the optimal reference solution with full knowledge of the future, giving a 2 – 10% higher cost, depending on the case at hand. These results include cases with forecast errors of over 30%, which indicates that our control approach is very robust. A closer look shows that there are still cases where the control performance can be further improved, for example at times with steep price changes. Also, the computational efficiency of the search algorithm should be improved, although it is sufficient for practical use at a limited (neighbourhood-level) scale. In a decentralized context, the same holds for the communication overhead. However, we should note that the robustness of the outcome allows us to use the algorithm less often than other methods such as e.g. the algorithms from Chapter 3 (e.g. 1 – 2 times per day). Note that a year simulation as in Section 3.6 is considered infeasible at the moment: we currently only have an agent implementation for the heating system, and extrapolating the simulation times in this chapter we expect a year simulation to take 7 weeks (although many performance improvement opportunities have not been exploited yet).

In our experiments, we found that the choice of bidding strategy for the devices is important for the cost outcome that we can achieve. An inappropriate bidding strategy for a device leads to poor control decisions, which may increase cost. We think that some devices can benefit from a device-level forecast, which it can use to better estimate its urgency.

Although this chapter has focused on a Walrasian auction dispatch mechanism as it is the most prevalent in the DSM domain, we think that the concept is in principle general enough to be adapted to other dispatch mechanisms. For example, if we take the house cost control mechanism of Section 2.4.2.1, then we could determine a strategy to set the commodity prices of the exchanger devices in this model.

Concluding, we think that the developed approach is a very flexible and robust control approach with competitive objective performance and various opportunities for further research.

4.6.1. RECOMMENDATIONS

The work presented in this chapter has lead to many ideas for future work. In the following, we give an overview of possible directions.

4.6.1.1. SEARCH IMPROVEMENTS

- 4.6.1.1.1. *Priority as variable in strategy* The experiments presented in Section 4.5 show that the search algorithm often has to use exaggerated supply quantities to get the desired response. Based on the results, we believe that it may be more effective to move the priority levels instead of the supply offer quantities in the simplified supply offer strategy representation. This should allow the use of less extreme steps in the case that the relevant priority range is small, and may increase the robustness of the supply offer strategy as we propose moderate supply offer values.
- 4.6.1.1.2. *Heuristics* We have used metaheuristics to address our problem. However, we think that the problem has sufficient (implicit) structure for the successful use of more problem specific heuristics. These specific heuristics may significantly speed up the search. For example, we know that a higher electricity price should correspond with lower supply offer values. Similarly, if we observe the use of an auxiliary resistance heating rod, this normally implies that the supply curve should have been higher in an earlier time interval to avoid the use of the heating rod. The latter may be used as a repair method during simulation (i.e. backtrack one time interval, increase the supply curve, go back further if necessary), or cancel search iterations that give poor results. Similarly, if we e.g. have washing machines or other devices with slow dynamics, then we may also consider to lower the demand in earlier time intervals to avoid a demand peak. With this domain specific knowledge, we may propose changes to the supply offer curve, and then use the simulation approach to check whether the change indeed works out as expected.

4.6.1.1.3. *Adaptive time and priority scale* The current search procedure uses the same time and priority scale over the entire horizon, for all search iterations. We expect to gain a significantly faster convergence if we can adapt the scale during the search. By this, we first find an approximate solution in the first iterations, which is then further refined in later iterations. Extra priority levels can be added during the search, although re-interpolation of existing solutions (e.g. when going from 3 to 4 equidistant levels) may decrease solution quality. Analog to the number of priority levels on the supply curve, extra time points can be added during the search. In a rolling horizon approach, we may even go further and choose to keep the part of the horizon that we do not use in execution at a low resolution. The assignment of more accurate values for the later time intervals is deferred to later planning sessions.

4.6.1.2. COMPUTATION AND COMMUNICATION COST IMPROVEMENTS

4.6.1.2.1. *Auction approximation* A straightforward decentralized implementation of the presented approach is inefficient: in principle, we have to communicate the bidding curve of each device $n_k \cdot n_t$ times in a planning session (we can avoid a factor $n_{\bar{e}}$ by multiplexing scenarios). As an alternative, we may iteratively approximate the outcome of the clearing process, e.g. by distributing an estimated clearing priority outcome vector. By this, we do not have to communicate separately for every time interval in every planning iteration. We expect that we can obtain a reasonably accurate estimate of the auction clearing process from which we can determine the estimated cost in 3 – 4 iterations (distribute the expected priorities, receive all bidding curves, clear markets, update priorities), and expect that previous planning iterations may improve this by giving a reasonable start approximation. Furthermore, we may invest more iterations in promising solutions. Note that if we choose to use supply offer curves that determine a fixed clearing priority outcome, then we may simply distribute the priority (i.e. we need only need a single iteration as no device can influence the clearing priority).

4.6.1.2.2. *Parallel evaluation* A different approach to reduce communication latency is to try n_{par} possible values of $\hat{\mathbf{x}}$ in parallel. However, this is wasteful if the probability to find an improving solution is high (i.e. in the first 30 iterations), as the parallel tracks can only exploit the information from the improving solution in the next step. To counter this, we may adapt n_{par} as the search proceeds. In a centralized implementation, this approach may also be used to perform the search in parallel.

4.6.1.2.3. *Fitness approximation* In a decentralized context, the evaluation of the fitness function induces high communication cost. Literature on metaheuristic optimization proposes fitness approximation to reduce the cost of evaluating the fitness function. For this, the evolution process uses an estimation function, and sparingly uses the more expensive “real” fitness function. The estimation function may be a neural network or a more structured representation such as a lumped model as in Section 2.3.2.1.5, except that we use simulations instead of historical data to estimate the available flexibility. The estimation may for example also first try an assignment for a small set of tracer devices as in [166], and then use the more detailed analysis to validate the most promising assignments.

4.6.1.3. FEATURE EXTENSIONS

4.6.1.3.1. *Dynamic prices* The current approach assumes that the prices over the optimization horizon are known in advance, and that the control policy does not directly depend on the prices. In future work, it may be possible to add variables to the optimization problem that introduce a dependency of the supply offer values on the current price, e.g. with a linear coefficient. As a positive side effect, this also decouples the resolution of the control policy set points from the resolution of price changes: to adapt to a different price, the policy does not need to change. With the presented optimization approach, this information can in principle be incorporated straightforwardly as an input parameter. However, we have not studied which information should be used, and how this information should modify the supply curve. In this case, the optimization may determine the mapping of the information to the way that the supply curve is modified, at the cost of increasing the problem size.

4.6.1.3.2. *Curve transformation* In this chapter, we have presented a case where the aggregator controls the supply of electricity with a supply offer agent. There are also cases where the supply is given e.g. by a generator, and the aggregator instead controls a supply or bidding curve transformation. For example, Kok [195: Chapter 7] transforms the aggregate demand bidding curve to account for capacity constraints. More generally, a similar approach to what we have presented in this chapter may be used to control the bias and stretching of the original supply offer and demand bidding curves. The transformation may work on the level of individual bidding and offer functions, but also on the aggregate bidding and offer functions.

4.6.1.4. ALTERNATIVE APPROACHES

4.6.1.4.1. *Bidding and supply curve optimization* In our experiments, we assume that the device bidding functions are given on beforehand. If we also can choose the bidding strategies of the devices, then we may “train” the devices to conform to the optimal solution that we find with a more conventional optimization approach (e.g. a QP solution). The simulation traces give information on what kind of bid should have been offered in different situations (i.e. current SoC levels) to give the desired response in each of the simulated scenarios. By fitting the supply curves and/or the bidding curves to a given set of optimal solutions, a supply offer strategy and corresponding device bidding strategies can be derived, both for the central control and for the individual device control. Note that this approach is comparable to the work in [246], although the approach we propose here accounts for the behaviour of the dispatch process.

4.6.1.4.2. *Dispatch perturbation* The (auction) dispatch simulation allows us to “measure” the error of the dispatch mechanism in comparison with the reference optimal control. We may use this error estimate to revise the planning, for example by adding the error as an extra demand vector during the planning with a more conventional approach. This estimation should be performed iteratively, but we expect 2 – 3 iterations to be sufficient. This type of analysis may be considered with multiple

scenarios. Note that in the use case that we discussed in this chapter (Section 4.4), this would mean that if we induce a high demand peak due to the use of the heating rod, then we do not address the high demand of the heating rod itself, but instead leave a “gap” in the aggregate level planning to cope with this demand. The error may also be considered at a more detailed level, and give extra constraints for the devices. A similar approach may also be used to compensate for other second-order effects, such as transport and storage losses.

- 4.6.1.4.3. *Co-design of planning and dispatch* In this chapter, we have focused on the use of an auction dispatch mechanism with general bidding functions for the devices. However, this mechanism clearly introduces high complexity in the planning phase, because it relates all device decisions with limited structure. Therefore, we think that it is worthwhile to consider a *co-design* of the planning strategy and the dispatch strategy. If the environment imposes the use of an auction dispatch mechanism, then we may choose the device bidding functions in such a way that these exhibit behaviour that is easy to model in an optimization problem. Preliminary work that considers the auction directly as an optimization problem and combines the bidding strategy of [265] with simulation based feedback shows promising results for an efficient and effective coordination.

COORDINATION IN PRACTICE

5

Abstract

There are some practical barriers for the uptake of DSM at a household level. One of these barriers is a lack of standardization of the interface to flexible devices, leading to high software development and maintenance costs. Control methods differ in their perspective on flexibility, which makes it difficult to find an useful common ground. Therefore, the *energy flexibility interface* (EFI) proposes to communicate the structure of energy flexibility instead of a specific perspective on flexibility. We develop a comprehensive TRIANA energy application prototype that implements the EFI. The prototype supports the decentralized planning and control of real devices on low cost embedded hardware, and demonstrates that the concepts of this thesis are applicable in an externally given framework, even though it can not yet be seen as a complete solution. It also shows that EFI maps to multiple perspectives on energy flexibility in addition to just-in-time auction based methods.

EFI/EF-PI 193 · TRIANA–EF-PI 198 · control space adapters 203 · experiments 219

5.1. INTRODUCTION

BESIDES CHALLENGES to orchestrate large groups of devices, DSM also gives more down-to-earth “last mile” problems to actually use devices that pose noteworthy challenges which are not always visible at the level of an abstract model. We may figuratively say that practice adds an extra dimension to the DSM problem. In this chapter, we develop a more practical implementation of the TRIANA DSM approach based on the optimization techniques from Chapter 3, which allows us to explore the applicability of TRIANA in a more realistic environment than only simulation. This environment helps to validate the proposed methods.

During the development and application of the PowerMatcher [195] DSM approach for real-world applications, the involved researchers found that the development of specific control agents for every possible device type is hard to realize in practice. We have a similar experience in field tests within the research group at the University of Twente [20, 362], where tailored optimization and driver software are

↪ Parts of this chapter have been published in:

[^HT:5] “Energy management with TRIANA on FPAI”. In *PowerTech*. IEEE PES, JUN 2015. DOI 10.1109/PTC.2015.7232650.

[68] DAGIOGLOU Stergios. “An implementation of the planning phase of TRIANA using the Flexible Power Application Infrastructure”. MSc thesis. University of Twente, AUG 2014. <http://purl.utwente.nl/essays/70985> (⌚ SEP 19, 2016).

developed to support the specific devices that are used within a project. Therefore, we need an abstraction of the features and capabilities of concrete devices, and even with such an (internal) abstraction the development and maintenance of the corresponding device drivers tend to take up the bulk of the engineering effort for an energy management application. This trend also exists in the world of operating systems: for example, the Linux kernel consists for 70% out of drivers [58],¹ and we see comparable figures in our DSM software implementations (e.g. 77% for the work in this chapter). Still, every operating system offers a (slightly) different device abstraction, and thereby has to replicate the effort to support a large number of devices. A standardized device abstraction is desirable to share development effort, but naturally raises the question where to standardize on. Energy management applications have similar problems. Different energy management approaches have different perspectives on devices, which may lead to wide, unwieldy interfaces, and thereby to a large driver development effort. The effort in agent software development has been acknowledged in standards development by providing versions of standards with limited features that can be implemented more easily [252]. We think that these lite standards are an undesirable development, because these unnecessarily limit the available flexibility. A well designed interface may avoid much of this effort, expose most of the available flexibility, and allow for the sharing of drivers between DSM approaches.

Instead of developing specific agents for every possible device type, the Power-Matcher researchers have proposed the *energy flexibility interface* (EFI). The EFI is an abstraction approach that proposes to let device drivers expose the *structure* of the energy flexibility. This structure is divided in flexibility classes. Subsequently, an energy management approach can address structurally similar devices (of the same class) with a single agent. By this, we may develop relatively simple device drivers, which are supported in the energy management application by relatively complex agents that can address the optimization of a complete flexibility class (or a relevant subset of this class). This asks for a large up-front investment to be able to optimize these more complex problems, but may then support a large class of devices with no further effort.

The EFI is implemented in the *Energy Flexibility Platform and interface* (EF-Pi). EF-Pi is a modular Java/OSGi-based platform for the development of energy applications. EF-Pi handles various peripheral concerns that are not directly related to energy management, such as the configuration of device connections and the UI.

Although EF-Pi and EFI were *designed* to accommodate multiple energy management approaches, until now the platform has been demonstrated only in combination with PowerMatcher. Dagioglou [68] developed a small proof-of-concept implementation of TRIANA for an early version of EF-Pi (which was at the time still called FPAI). The promising results of this proof-of-concept have lead us to develop a more advanced implementation for the current version of EF-Pi, with more complete support for the platform and its energy flexibility interface.

1. With the rise of popularity of Linux in embedded systems, this has risen to 77.8% in Linux 4.6, counting the drivers and arch directories with SLOccount [353].

In this chapter, we elaborate on this design and the implementation of TRIANA on the EF-Pi platform, and the challenges that we addressed to turn the concepts of Chapter 3 into a fairly complete DSM implementation that can optimize the control over real devices. Note that the experience acquired during the development of TRIANA-EF-Pi has resulted in several extensions to the techniques in Chapter 3, in particular relating to the use of nonuniform time interval lengths, using planning for control, and multicommodity/hybrid energy optimization.

This chapter is structured as follows. First, we present some background on energy management standardization and on EFI/EF-Pi in Section 5.2. We continue with the realization of TRIANA-EF-Pi in Section 5.3. The *control space adapters*, which intermediate between the EFI flexibility classes and TRIANA, are an important part of this realization and are discussed in Section 5.4. We assess the performance of these control space adapters in Section 5.5, and end with conclusions in Section 5.6.

5.2. EF-PI: A FLEXIBLE ENERGY MANAGEMENT PLATFORM

5.2.1. THE INTEROPERABILITY CHALLENGE

The integration of multiple, separately designed systems is often an ordeal. Usually, support for integration with other (in-building) systems is limited or not available. This leads to a *silo effect*: devices can only cooperate with devices of the same (or a “befriended”) vendor. Clearly, this aspect makes coordination a lot more difficult unless one is willing to buy everything from a single vendor, and may rely either on politeness and guesswork of the involved parties, on an overarching coordinator, or on common standards for cooperation.

Demand side management concerns control over the demand, i.e. over devices. Home automation is a popular field that aims for this control as well, but mostly focuses on aspects unrelated to energy management. A large body of software platforms and standards have emerged for home automation, e.g. openHAB [254] and ZigBee Home Automation [379]. A recent overview of the most prevalent smart home platforms in Germany can be found in [47].

Independently of the home automation domain, standards have also been developed for energy management, e.g. OpenADR [253]. Furthermore, several recent home automation standards incorporate energy control features as well [380, 382]. These standards focus on energy flexibility from the perspective of an electric utility (e.g. with billing information and DR commands), and appear to pay less attention to how this maps to real devices. Residential standards focus on demand response with *heating, ventilation and air conditioning* (HVAC) systems, because these represent the bulk of the controllable load, especially in the US. In larger buildings, building energy management systems are common, which also focus on HVAC. These management systems in general have (vendor-)specific optimization solutions, e.g. [366].

In the near future, smart grid compatibility might become a requirement or a “must-have” feature for new appliances. Consequently, the major appliance manufacturers are scrambling to become “smart grid ready”. As there is no agreement

FIGURE 5.1: Example $n \times m$ compiler problem (not necessarily representative of the real world). Each edge represents one compiler/translator.

on standards yet (and even not on what *readiness* covers), every manufacturer fills this concept with their own definition and proprietary interfaces. In practice, this readiness ranges from providing a manual remote control interface to full-blown economic optimization systems. Large, expensive devices tend to have more extensive controllers than small devices. However, manufacturers give only limited external access to these controllers to protect the resource.

To support this sea of possible devices, DSM software needs a unified view to access these devices. DSM software typically provides an *application programming interface* (API) with which a developer can add support for a specific device to this specific DSM software implementation. These APIs tend to gear towards the concepts of the DSM approach, and not to those of the device manufacturer. Consequently, each of the DSM approaches needs a tailored driver implementation. Even worse, APIs are subject to change over time, so the implementation needs to be updated perpetually in order to work with the latest versions of the software. While this perpetual upgrade model more-or-less succeeds in the world of desktop software, the embedded world is a lot more conservative: with few exceptions, once a product is delivered, only highly critical, device- or reputation-damaging problems are resolved by software updates. It is therefore unlikely that device manufacturers will keep up with the pace of DSM development. In practice, this means that the integration and maintenance tasks are shifted from the device manufacturer to the DSM software developer. As the development resources of the DSM software developer are limited as well, current DSM platforms offer a patchwork of support for the smart devices that are currently popular on the market. It should be clear that this model of software development is not sustainable in terms of engineering effort, especially as the number of smart device models increases. As a consequence, the developers of DSM software and devices should cooperate and introduce a standard for device flexibility that serves the needs of both involved parties.

This maintainability problem is analogous to the $n \times m$ problem, which is well known from the context of compiler construction [349: pp. 40–42], and is illustrated in Figure 5.1. Consider a world with n programming languages and

m target machines. A compiler translates a program written in a language x to an object file for a target y . We need a specific compiler for every combination of $x \rightarrow y$, even though most programming languages and target machines are conceptually highly similar. Rather than diligently building $n \times m$ compilers (or just a popular subset of these), clever designers introduce an *intermediate representation* (IR). This intermediate representation abstracts the differences between the machines. The resulting new problem is to build translators from each of the n languages to the “abstract machine” offered by the IR, and a family of translators from the IR to each of the m target machines. An example of an implementation of an IR is LLVM [203]. This approach turns the $n \times m$ problem into an $n + 1 + m$ problem, where 1 represents the effort relating to the IR itself. Unless n and m are very small (≤ 2), this approach pays off quickly, even when the effort to develop and target the IR is high. However, note that it is typically lower, as a serious implementation of a compiler with state of the art optimization techniques may have a cost comparable to the development of these techniques for the IR. In the example of Figure 5.1, we go from a problem with 16 compilers to a problem with 8 compilers (however, as indicated in the example, only the most popular combinations may be implemented in practice; in the case of the example, 10 combinations are implemented). To make the analogy explicit, the programming languages correspond to devices, and the machines correspond to DSM approaches (or *energy applications* in the terminology of EF-Pi).

Although the IR approach has major theoretical and practical benefits, the main challenge of IRs is that information can get lost in translation to the IR. A possible consequence of this is that we lose efficiency. Consequently, components need to be sufficiently similar. The definition of the IR should be chosen carefully, such that it supports as many systems as possible, but it also has to remain useful for the systems that are supported. This raises the important question: what is the common language between devices and energy applications?

5.2.2. ENERGY FLEXIBILITY INTERFACE AND PLATFORM

The *energy flexibility interface* (EFI) aims to become the common language between devices and energy applications. Usually, standards focus mostly on either the devices (by exposing the available switches and measurement values) or on the energy applications (e.g. by delegating optimization problems to device drivers). Note that we implicitly also propose such an energy application centric driver approach in Chapter 3, and this also used to be the implicit approach of PowerMatcher. In contrast, EFI decouples these worlds and captures the structure of the available energy flexibility. An open source Java implementation of EFI, the *Energy Flexibility Platform and interface* (EF-Pi), is available online [118]. In this chapter, we describe the mapping and implementation of TRIANA (as described in Chapter 3) on the EF-Pi platform. In the following, we shortly describe EFI and EF-Pi, and continue with TRIANA-EF-Pi in Section 5.3.

5.2.2.1. ENERGY FLEXIBILITY INTERFACE

EFI distinguishes itself from other concepts by the approach it uses to decouple the DSM application from the appliance drivers. There are many similarities between different devices, but also many differences. Rather than trying to come up with a single common IR that fits for all devices only very poorly, EFI proposes to introduce *four* languages, each of which describes a specific class of flexibility. This contrasts with an approach where we abstract to specific classes of devices, such as washing machines, EVs and microCHPs, and so on. These flexibility classes fit conceptually between the DSM view of resources and the perspective of appliance developers. In the sense of flexibility, washing machines may be class-wise equal to EVs (both have to “run” within a certain time interval) and microCHPs may be equal to heat pumps (both describe a thermostatic control problem). Given a specific device, a device driver developer should identify the most suitable flexibility class, or *control space* class present in the available set of flexibility classes. For every device model (or model family covered by the same driver), this binding needs to be developed *once* to support all energy applications that implement EFI.

The decoupling using control space classes gives benefits to both DSM software developers and appliance manufacturers. DSM software developers now only have to address four different control space classes, rather than each of the specific device classes, or even device models. Therefore, DSM software developers can focus on improving their DSM software, rather than continually maintaining device support. The appliance manufacturer’s device driver developer only has to target a single interface, instead of a different interface for every DSM implementation (or even implementation *version*).

The number of control space classes is a tradeoff between the effort on the side of the device driver developers and on the side of the DSM software developers. Similarly, the complexity of the control spaces trades off between accuracy and development effort. EFI aims to strike a balance between these aspects. Note that in practice helper classes are available as a front-end to implement drivers for common device classes, e.g. washing machines.

Whereas DSM software often takes a discretized view of time, devices work in a world with continuous time. Within EFI it has been chosen to represent time as a continuous, run-length encoded quantity. Events and intervals between events characterize the system behaviour. However, looking at the world of optimization, many algorithms assume that time is a discrete quantity. Using such optimization algorithms in DSM software therefore asks for a translation from continuous to discrete time, as well as the converse translation. This translation is inevitable, yet often it is implicitly pushed into the device drivers. EFI determines that this responsibility belongs within the DSM software.

To improve compatibility at the interfaces of components, EFI uses a unit type system to enforce that communicated values have the same meaning. This means that a driver that internally uses imperial (i.e. US) units can safely connect to DSM software that uses SI units without manual (error-prone) intervention. The type

system also reasons about the matching of, and the relation between connected quantities; *power* and *energy* are incompatible quantities, yet *power* $\times$ *time* and *energy* are compatible.

EFI supports multicommodity (hybrid) energy streams. For example, a hybrid heat pump may choose between consuming gas and electricity to provide heating, and gives the energy application the freedom to decide how the heat pump should be operated, thereby taking into account dynamic (day-ahead) prices for both gas and electricity (which some energy suppliers are already experimenting with [270]). In time, we expect that this leads to an integrated management of commodities.

The EFI allows device drivers to specify the internal cost of most operations. For example, a microCHP may be subject to wear when it is turned on or off. To rationally make use of the switching flexibility, the costs from this wearing may be accounted for explicitly. By this, the energy management application may make a tradeoff between the optimization of the external energy consumption behaviour and the wear or e.g. discomfort cost of the local resources. Note that the appropriate relative scaling of these costs in the optimization model relies on a clear view on the relation between the considered costs, which we develop in Section 3.2.

5.2.2.2. CONTROL SPACE CLASSES

The control space classes are the central part of EFI. Each control space class defines a message protocol that describes the allocations that the energy application sends to a device (e.g. “turn on”), and the updates that come from the devices (“the device has turned on”). We do not aim nor attempt to give a complete overview of EFI in this work. The precise set of classes and semantics may be subject to change during development. For a more detailed and up-to-date description, please refer to the platform documentation [117].

The EFI considers the following four control space classes:

- *Uncontrolled* describes devices that offer almost no flexibility, e.g. TV sets, lighting and PV inverters. Despite the name that suggests that no control is possible at all, device drivers may selectively allow curtailment. For example, a PV inverter may discard excess production up to a specified range, or the lights may be dimmed at some times.
- *Time Shiftable* describes devices for which a “job” has to be executed with start times and deadlines, e.g. washing machines, dishwashers and washer-dryer combinations. Jobs can have multiple parts, each with a known demand profile, and time limits between parts. The processing of a part is uninteruptible.
- *Buffer* describes devices that can be characterized by a continuous state variable (i.e. a SoC, fill level, or temperature), a set of actuators, losses and demands that change the state variable. EF-Pi always refers to the state variable as the fill level. An actuator can have multiple modes. A mode describes the behaviour of the actuator as a piecewise constant, partial function of

the state variable. System descriptions can define a minimum time distance between certain mode changes. EF-Pi describes these time constraints with mode transition lockout timers. A leakage function describes loss over time. Prominent examples of *Buffers* are batteries and thermal systems with storage.

- *Unconstrained* devices are in principle *Buffers* without the continuous state variable, for example a diesel generator. These devices have fewer restrictions than *Buffer* devices, and thereby allow specific optimizations.

Some devices may fit to multiple control space classes (e.g. EVs). In that case, the EF-Pi device driver developer should pick the class he assumes to be most appropriate.

5.2.2.3. ENERGY FLEXIBILITY PLATFORM

EF-Pi is a modular and open Java/OSGi-based platform for energy management applications that hosts EFI. EF-Pi presents EFI as a collection of immutable Java classes (work is underway to develop an XML representation of EFI). In addition to EFI, EF-Pi provides helper classes to e.g. define a washing machine in terms of the *Time Shiftable* control space class. The platform handles various services that do not directly contribute to energy management, such as configuration management and the (framework for the) user interface. Figure 5.2 shows a screenshot of the interface where EF-Pi runs the PowerMatcher energy management application and some simulated devices. A demo package with this configuration may be downloaded from [264]. A video introduction to EF-Pi is available at [5].

Figure 5.3 gives a screenshot from the configuration management interface for the connections between devices and energy applications, which allows for the drag-and-drop configuration of the system. The screenshot shows a two-level configuration of TRIANA with a (simulated) washing machine, a battery, an uncontrolled load and a generator. Based on the abstraction of EFI, devices can be dynamically connected to an energy application. We have demonstrated with a real washing machine that it is feasible to dynamically hand over the control of a washing machine to a different energy application, i.e. from PowerMatcher to TRIANA, by dragging the “controller” connection in this *graphical UI* (GUI). This is a large step in terms of interoperability between DSM approaches, which is made possible by EFI.

We introduce TRIANA–EF-Pi in the following section (Section 5.3). After this, we focus on the *control space adapters* in Section 5.4, which are needed to connect each of the control space classes to the optimization model of TRIANA. Each of these control space adapters implements a local (device level) optimization for the corresponding EFI control space class.

5.3. TRIANA–EF-PI

5.3.1. INTRODUCTION

To bring TRIANA to EF-Pi, we have developed a new implementation of TRIANA that incorporates the concepts of Chapter 3. In this section, we give a brief overview of this implementation.

FIGURE 5.2: Screenshot of EF-Pi with PowerMatcher energy application.

FIGURE 5.3: Screenshot of connection manager interface with a nested TRIANA-EF-Pi configuration and several devices.

The simulation tools for TRIANA (written in C++ and Python) have in principle not been designed to be used in a real implementation context. In order not to burden this simulation environment with details needed for a real-world prototype implementation, we choose to implement TRIANA separate from the simulation. Although there exists a different implementation of TRIANA that is targeted to real applications [20, 362], this implementation embedded in a proprietary .NET-based smart home product and thereby hard to integrate elsewhere. To avoid integration issues, we have ported TRIANA to Java. In this way, the EF-Pi platform and the DSM approach live in the same environment. By this, all relevant components of the platform can be managed by OSGi, which makes the deployment less complex.

5.3.1.1. STRUCTURE

Analogous to PowerMatcher on EF-Pi, TRIANA is split up into a core implementation (TRIANA itself) and a platform-specific adapter (TRIANA-EF-Pi). The implementation of TRIANA is decoupled from EF-Pi, and works without EF-Pi and OSGi (it is possible to start directly from main). TRIANA depends only on the Java standard library. By this, we may also use the Java implementation of TRIANA with different Java-based platforms (with a suitable adapter). To use TRIANA, it should be embedded into *some* environment with a portable lightweight environment adapter, which describes e.g. how we should write a logging entry or schedule a task.

Although there are many similarities between the set of devices that are supported by energy application platforms, the exact definition of the semantics of devices varies widely, and a shared model seems impractical at this moment. If we would target multiple platforms, then we have to be careful to keep the meaning of the optimization model consistent with each of the platforms that it supports. Consequently, for now we put the optimization software for the control space classes within the platform adapter. In this way, we should be able to target the port to other (Java-based) energy application platforms and to multiple versions of EF-Pi, with limited risk of interference. A conceptually better approach may be to introduce internal abstractions to express the problems of each platform within a common optimization model. We present the device-level optimization problems (one for each control space class) in Section 5.4.

5.3.1.2. CONTROLLER MANAGER

The EF-Pi *controller manager* is the main entry point of an energy application. We illustrate in Figure 5.4 how TRIANA implements this entry point, and how it relates to the other components. The controller manager embeds a household level internal aggregator, which corresponds with a group (internal) node from Section 3.2. The aggregator embeds the IDDP and profile steering group optimization algorithms (column generation has not yet been implemented in this context).

The EFI devices are connected to TRIANA through control space adapters (which are supplied by TRIANA-EF-Pi). The internal aggregator contains a server to connect the control space adapters, and the control space adapters contain a

FIGURE 5.4: Example instance of deployment of TRIANA-EF-Pi on a household level. The grey nodes are part of TRIANA, the hatched nodes belong to TRIANA-EF-Pi, the dotted nodes correspond to device drivers, and the white nodes are part of EF-Pi (upper part) or the devices below it (lower part).

client to interact with this server. The controller manager allows EFI devices connect to TRIANA. When a device is connected to the TRIANA energy application in EF-Pi, TRIANA-EF-Pi instantiates a control space adapter of the appropriate type, which connects to the internal aggregator. A control space adapter accepts EFI control space updates and sends EFI control space allocations of the corresponding control space class through an object-based “connection” to the EF-Pi driver of the device,² which then communicates with the controller that is embedded in the actual device (e.g. over Modbus, ZigBee, or some IP-based method). An internal aggregator also contains a client, which allows aggregators to be nested, as already shown in Figure 5.3. Aggregator nodes and a device (control space adapter) nodes have the same upward interface. We may specify an URI address to connect to the server of a TRIANA instance outside of the considered EF-Pi instance, e.g. at a neighbourhood level.

A client is an active component that can take the control role of a root node in TRIANA, and may be used to implement a local control over the subtree in the case of a connection failure (this has only been implemented partly). The root node should obtain some optimization goal (an imposed pricing scheme). Although it is possible to define a special node above the root node to specify the goal (e.g. to integrate TRIANA in the control system of a DSO or electricity retailer), for now we have hard-coded this to either a balancing objective or a price optimization objective, depending on the case at hand. The client is the source of planning re-

2. In principle, EF-Pi further splits up a device driver into a *resource manager* and a *resource driver*, although this separation is often ignored in practice.

quests for the resource below it (either a group planner or a local device planner). In the TRIANA hierarchy, Pricing objects go downwards (describing an upward cost function), and Pattern objects go upward (describing the demand pattern for each commodity and the internal cost of a planning).

5.3.1.3. COMMUNICATION ABSTRACTION

To enable decentralized control over multiple systems, we have developed a communication abstraction layer that allows the nodes of TRIANA to communicate through different media. The communication layer defines the server side (to accept connections from children in an aggregator) and the client side (to connect to an aggregator). We apply this communication abstraction layer e.g. in Appendix C.3.1 for a Protocol Buffers [139] based protocol, and within the e-balance research project for a publish/subscribe middleware. Furthermore, there is an implementation that passes Java objects directly within the same EF-Pi instance (e.g. between the control space adapters and the household internal aggregator). The communication abstraction is configured with an URI-based scheme.

5.3.1.4. DISCRETIZED TIME

The EFI introduces a continuous (run length encoded) time base within its energy descriptions. Although many concepts of TRIANA may be translated to a system with continuous time, for the sake of efficiency and simplicity we choose to work with discretized time, and perform appropriate marshalling between the worlds of continuous and discrete time. This means that real-time stamps are rounded up or down to a full time interval (according to the context), and that discrete time solutions are translated back to continuous time.

To improve the time granularity of control while keeping the total number of time intervals limited, our implementation uses time intervals of nonuniform length, and make the time base of profiles explicit (i.e. a start time and a list of time interval lengths). We may in principle combine profiles of different time bases by splitting intervals at the points where the profiles do not agree. However, this often leads to too many splittings, which may mean that we end up with time intervals of the lowest granularity (in our case at the level of seconds). Therefore, we impose the used (nonuniform) time base from above. The root node may choose its own time base. A common time base between nodes greatly simplifies the implementation of the group optimization algorithms (although it can still be nonuniform over time).

5.3.1.5. MULTICOMMODITY

As we have already discussed in Section 3.2.4, we can naturally extend TRIANA with multicommodity optimization support. Although EFI currently lumps all demand of a specific commodity type within a household, we generalize the concept of commodities to commodity ports: there can be more than a single connection of some commodity, with different cost for each port (think of a three-phase electricity connection that may be balanced by connecting a device to the appropriate phase). As we have already described in Section 3.2.4, TRIANA can address multicommodity problems at a group level in a way similar to how it handles multiple time intervals.

The extra dimension adds a linear cost in time to the cost computation of the local algorithms. Multicommodity devices offer more flexibility (more operating modes), which usually adds a linear or quadratic increase in complexity to the local algorithms. The number of iterations at the group level may increase as well. If the number of commodities is small (e.g. electricity, gas, and heat), the extra cost for scheduling is limited.

5.3.1.6. PLANNING SESSIONS

Whereas a simulation can sequentially perform the planning and then apply the result on this planning afterwards, this is less straightforward in a real, decentralized implementation. We do not want the system to already start performing control actions based on the first iterations of a planning session. To distinguish between planning requests, we attach a unique plan identifier to each planning (candidate pricing scheme) request and pattern response. The concept of a planning session is introduced only within the group planner. When the planning session is finished, it may choose to *commit* to a certain pricing scheme, which instructs each of the children to use an earlier given planning according to its plan identifier. The commit instruction of TRIANA lets each of the control space adapters send a control space allocation to its connected device that corresponds with the planning. Note that the planning result, the TRIANA pattern and the control space allocation each give a slightly different perspective on the problem (e.g. for a washing machine, this may respectively give “start the job in time interval 5”, a demand pattern that has a demand block from 17:00–19:00 and “start the *Time Shiftable* at 17:00”).

5.4. TRIANA-EF-PI CONTROL SPACE ADAPTERS

Energy applications in EF-Pi have to provide a “driver” for each of the control space classes, which translates the EFI abstraction into concepts that are meaningful within the energy application (e.g. demand profiles or bidding curves). We refer to these “drivers” as *control space adapters* (CSAs). These control space adapters are the converse of device drivers (which express the device capabilities in terms of EFI). We find that the development of effective control space adapters poses one of the main challenges in the porting of an energy application to the EF-Pi platform.

A control space adapter presents a controllable resource as a TRIANA client. The main challenge in each control space adapter is to solve a local (device-level) node optimization problem. In the following, we present a corresponding scheduling approach for each of the control space classes. For the *Uncontrolled* class, we may use a greedy algorithm as it does not have dependencies over time. The *Time Shiftable* CSA and *Buffer* CSA for TRIANA use a specialized DP formulation for their local optimization problem (see [51] for background information on dynamic programming). The *Buffer* has a quite complex structure and is thereby less straightforward to optimize. The *Unconstrained* class is a subset of the *Buffer* class, and can thereby use a reduced version of the DP formulation that we developed for *Buffers*.

5.4.1. UNCONTROLLED CSA

The *Uncontrolled* control space is a simple class that in principle only describes the demand of a device, but may also offer the ability to curtail it to a certain range at given times. By this, we may for example limit the output of a PV inverter. The planning of an *Uncontrolled* device determines the expected demand over time. To this end, the control space adapter uses the forecast which should be provided by the *Uncontrolled* resource manager. In practice, many of these managers do not provide a forecast. In this case, we derive a simplified forecast from the current measurement. The current measurement subsumes the forecast in the very short term. We integrate the current measurement with an exponential decay factor. We used this *Uncontrollable* control space adapter with a real PV inverter and with a real smart meter.

Support for curtailment has not yet been implemented, but can be added trivially when needed. The corresponding optimization problem has no state, and may thereby be considered as a degenerate DP. As a result, the problem degenerates to a greedy local selection of the minimum cost. For most pricing structures (e.g. linear, or quadratic functions), this selection is trivial: for linear pricing it means picking the minimum or the maximum permitted curtailment value, depending on whether the price is positive or negative. For quadratic pricing, the CSA picks the feasible solution closest to the minimum of the quadratic function for each commodity.

5.4.2. TIME SHIFTABLE CSA

5.4.2.1. DISCRETIZATION

The *Time Shiftable* control space describes a sequence of static demand patterns (segments) with constraints on the start times at which these patterns can be started. As the first step, we discretize the values in the control space: we round up the arrival time of a job, and round down the deadline. For a device i and a job j , this gives a discretized job arrival time $t_{i,j}^a$ and deadline $t_{i,j}^d$. A job consists of multiple segments s , which have a demand profile $\mathbf{x}_{i,j,s}$ and a length $t_{i,j,s}^l$. The planning assigns a start time $t_{i,j,s}^b$ to each segment in a job. Implicitly, this also gives the completion time $t_{i,j,s}^c (= t_{i,j,s}^b + t_{i,j,s}^l)$. Figure 5.5 illustrates the discretization of the *Time Shiftable* control space for the DP.

5.4.2.2. STRAIGHTFORWARD DP APPROACH

To solve the corresponding optimization problem, we first use a straightforward DP formulation. We represent states with pairs $\langle t, s \rangle$, where $t \in \{1, \dots, n_t\}$ is the time interval and s is the number of the current segment. We assume that the planning has to account for at most a single job j . As is usual in DPs, we assume that the commodity cost function is separable between time intervals.

The planning has to account for the maximum off-time between segments. This accounting may be implemented by adding a timer variable to the state. However, this adds complexity and increases the size of the state space. Instead, we avoid

FIGURE 5.5: Example of a *Time Shiftable* job description and the discretization of this description, including the demand profile for a single commodity.

FIGURE 5.6: Jumps between time intervals in the *Time Shiftable* DP corresponding to Figure 5.5, with example commodity costs.

representing these timer variables by exploring all possible off-time values that are feasible in the current state. For each “jump” value (i.e. a possible discrete number of time intervals that we can leave the device off), we let the next segment start immediately, except at the end of a job. Figure 5.6 demonstrates these jumps with arrows ($\uparrow$). Note that while this avoids the space complexity to represent the timers, the jumps do not avoid an increase in time complexity, which is linear in the length of the maximum off-time (in intervals). We can avoid this increase in time complexity by sampling the off-time interval a fixed number of positions (e.g. the beginning, the middle, and the end), at the expense of possible suboptimality.

For a specific *Time Shiftable* i and job j , the DP solves the following optimization problem:

$$\min z_{\text{DP}}(t_{i,j,s}^a, 1) \quad (5.1)$$

$$z_{\text{DP}}(t, s) = \begin{cases} 0 & \text{if } t > n_t \\ \sum_{t' \in \{1, \dots, t_{i,j,s}^l\}} z'(x_{i,j,s}(t'), t + t' - 1) + z_{\text{DP}}(t + t_{i,j,s}^l, s + 1) & \text{if } t_{i,j,s}^b = t \text{ and } 1 \leq s \leq n_{s,i,j} \\ z(0, t) + z_{\text{DP}}(t + 1, s) & \text{otherwise} \end{cases} \quad (5.2)$$

$$z'(x, t) = \begin{cases} z(x, t) & \text{if } 1 \leq t \leq n_t \\ 0 & \text{otherwise} \end{cases} \quad (5.3)$$

$$t_{i,j,s}^l = |\mathbf{x}_{i,j,s}| \quad \forall s \in \{1, \dots, n_{s,i,j}\} \quad (5.4)$$

$$t_{i,j,s}^c = t_{i,j,s}^b + t_{i,j,s}^l \quad \forall s \in \{1, \dots, n_{s,i,j}\} \quad (5.5)$$

$$t_{i,j,1}^b \geq t_{i,j}^a \quad (5.6)$$

$$t_{i,j,n_{s,i,j}}^c \leq t_{i,j}^d \quad (5.7)$$

$$t_{i,j,s-1}^c \leq t_{i,j,s}^b \leq t_{i,j,s-1}^c + t_{i,j,s}^{\text{gap}} \quad \forall s \in \{2, \dots, n_{s,i,j}\} \quad (5.8)$$

For notation convenience, the description assumes a common time interval length τ . Similarly, while the presentation suggests that the segment demand vector $\mathbf{x}_{i,j,s}$ contains scalar values for each time interval, we can (and do) use the same structure with vector values to solve multicommodity problems.

5.4.2.3. ZERO DEMAND GIVES ZERO COST

For most cost functions, it holds that if the demand is 0, then the cost is 0 as well. In this case, we do not have to account for the cost in time intervals that do not perform work, which slightly simplifies the DP implementation. Otherwise, we can often add a suitable bias term to the cost function such that $z(0, t) = 0$: for example, $z(x, t) = \tau(x - p_t)^2 = \tau x^2 - 2\tau p_t x + \tau p_t^2$, i.e. the profile steering cost function of (3.66), has the bias term $-\tau p_t^2$. As the bias terms are equal in every solution, these terms may subsequently be ignored in the optimization. However, for validation purposes, it is good practice to still add the bias terms afterwards. The bias term can be computed outside of the DP procedure, and thereby does not affect performance.

5.4.2.4. EQUAL COEFFICIENTS GIVE EQUAL COST

We do not account for the “profit” of spreading the demand over multiple time intervals. This is in particular inappropriate when there is a variable time scale, because this means that demand is cheaper in a long time interval than in a short time interval. If the quadratic costs are unequal between time intervals, then we may use the RMS values of the demand to account for the quadratic cost in a fair and accurate way (next to the averaged profile that is representative for linear cost, we may determine an RMS profile that is representative for the quadratic cost).

5.4.2.5. PROFILE DISCRETIZATION

To determine the cost to start segment j at time $t_{i,j,s}^b$, we have to evaluate the cost to run over the following time intervals, for the length of the segment profile. Depending on the time interval structure, this can become an expensive operation. We can evaluate the cost separate from the jump decision.

If the time interval length is fixed, then we can discretize of each segment demand profile once, and use it throughout the DP: the demand discretization is shift invariant, as long as the start time aligns with a time interval. We still need to determine the cost for every possible start time.

If the time interval length is variable, then we have to re-discretize the segment profile at every point where the time interval length changes within the scope of the segment profile time duration (usually $\mathcal{O}(n_t n_s)$ times, where n_s is the number of segments). This step makes the planning slow: the segment profile is often defined at a fine time scale, and is not aligned to the time base of the planning. This leads to an interpolation step, which is linear but expensive in practice.

A possible solution is to choose a very small time interval length, for example the *greatest common divisor* (GCD) of all time interval lengths (which ensures alignment at each time step), and to solve the problem with this fixed time interval length. However, a small time interval length makes the search expensive as well. The time complexity of the DP is in $\mathcal{O}(n_t n_s t^l + n_t n_s t^{\text{gap}})$, where t^l is the (highest) segment length. Therefore, as n_t , t^l and t^{gap} are all inversely proportional with the time interval length τ , the overall complexity is quadratic in τ^{-1} . If we consider only the time points that correspond with the original time base as possible segment start points, then the n_t term remains the same as before. Also, as proposed in Section 5.4.2.2, we can sample the range of t^{gap} at a fixed number of points to avoid the increase in complexity. However, we now still have the dependency of t^l on τ .

In the following, we consider an elegant alternative approach to minimize segment cost where the time complexity does not depend on t^l , but which does have extra constraints on the problem structure.

5.4.2.6. CONVOLUTION TRANSFORMATION

If the cost function only contains linear commodity prices (or can be reduced to this) and we use a single time interval length, then we can use the following alternative method to determine the cost to execute a *Time Shiftable* segment. Let $c(t)$ be the linear commodity price at time t . In this case, the simplified, direct cost function becomes (for fixed i and j and a single segment):

$$z(t_{i,j,s}^b) = \sum_{t \in [t_{i,j}^a, t_{i,j}^d)} c(t) x'(t - t_{i,j,s}^b + 1) \quad (5.9)$$

$$x'(t') = \begin{cases} x_{i,j,1}(t') & \text{if } 1 \leq t' \leq t_{i,j,1}^l \\ 0 & \text{otherwise} \end{cases} \quad (5.10)$$

Equation (5.9) is equivalent with the definition of cross-correlation [358]:

$$(f \star g)(n) = \sum_{m=-\infty}^{\infty} f^*(m)g(m+n), \quad (5.11)$$

if we take $f(m) \equiv c(t)$ (with $f^*(m) = f(m)$ for real m), $g(m+n) \equiv x'(m+n)$, and $n \equiv -t_{i,j,s}^b + 1$. Cross-correlation is almost the same as convolution:

$$(f \star g)(n) = \sum_{m=-\infty}^{\infty} f(m)g(m-n). \quad (5.12)$$

If we reverse either $\mathbf{c}$ or $\mathbf{x}'$, then the convolution performs the same operation as the cross-correlation, and thereby as the original cost function. A well-known result in signal processing is that (circular) convolution is equivalent to multiplication in the frequency domain:

$$(\mathcal{F}(f \star g))(\omega) = (\mathcal{F}(f) \cdot \mathcal{F}(g))(\omega) \quad (5.13)$$

Hereby, we can implement convolution by first transforming both f and g to the frequency domain with operator $\mathcal{F}$, then performing pairwise multiplications between $\mathcal{F}(f)$ and $\mathcal{F}(g)$, and then going back to the time domain with $\mathcal{F}^{-1}$. Both $\mathcal{F}$ and $\mathcal{F}^{-1}$ are usually implemented with the *fast Fourier transform* (FFT) and the (almost equal) *inverse FFT* (IFFT), which have a time complexity of $\mathcal{O}(n \log n)$, where n is the length of the input. Although we now have the *circular* convolution, the difference with the “normal” convolution can be ignored or easily circumvented in most practical cases. In our case, we can ignore the positions in the output array that use wrapped data, as these correspond with execution before the arrival time or after the deadline. To solve the original problem, we determine (the index of) the lowest value in the output array that does not violate the arrival and deadline constraints.

With the presented approach, we can minimize the commodity cost of a single segment. To support multiple segments, we can use this approach to precompute the cost to start each segment at a particular time, and subsequently solve the multiple segment problem with (a variant of) the original DP problem. The modified original problem now chooses a path through the start cost profiles that satisfies the *Time Shiftable* constraints.

While the presented approach gives a major theoretical improvement in performance (from n^2 to $n \log n$), the practical improvement is not as large in the context of our original problem as may be expected. To estimate the benefit from the complexity improvement, consider the following. To use the presented approach, we have to execute 3 FFTs with $t_{i,j,s}^d - t_{i,j,s}^a$ values (the number of intervals n). We assume that an FFT operation is 1.5 times as expensive per element as a direct convolution evaluation,³ which gives $4.5 n \log_2 n$ units of work. We assume that the length of the job is $1/10^{\text{th}}$ of the window length (e.g. 2 out of 20 hours for

3. We estimated this using NumPy by benchmarking a 256-value FFT (divided by $\log_2 256$ to factor out the log term) and a 256-value dot product, which is similar to a convolution in terms of operations.

a washing machine). For the direct approach, this means that we have to check $1 - 1/10 = 9/10^{\text{th}}$ of the possible start positions, which gives $1/10 \cdot 9/10 \cdot n^2 = 9/100 n^2$ units of work. Under these assumptions, the break even point where the FFT method becomes less expensive than the direct approach is at $n = 439$, i.e. with 3.3 minute time intervals on a 24-hour horizon. Given the long time duration of typical *Time Shiftable* jobs (and the lack of control after the job has started), it is questionable whether scalable support for increasingly small time granularities is really necessary for a practical implementation.

If the cost function has quadratic costs that vary over the considered optimization horizon, then we may account for this by adding an extra “commodity” dimension to the problem (if the costs are constant, then we already know from Section 5.4.2.4 that these costs can be ignored). In this dimension, we take the squared value of each power value in the demand profile, such that the optimization problem considers a convolution between the quadratic coefficients and the squared demand power values, which gives the intended result. Note that this approach applies more generally to cost functions where only the coefficients are time variant.

5.4.3. BUFFER CSA

The *Buffer* control space is a very flexible (but also complex) class that describes e.g. EVs, microCHPs and heat pumps with a common model. The class describes the response of a (dynamical) system that has a single SoC variable. There is a set of actuators that operate on the SoC (e.g. a heat pump and a heating rod). Each actuator runs in a certain mode, which is selected from a set of possible running modes. The system description describes how a running mode affects the SoC with a fill level function, which is a piecewise constant function that is dependent itself on the SoC. The combination of actuators fill level functions determines how the SoC changes over time, together with a description of the losses (which is also a fill level function). An actuator can switch between modes, and a graph describes the possible transitions between the running modes of an actuator. Actuators have discrete modes that describe the behaviour of the actuator with a fill level function. This fill level function is a behaviour description that includes the fill level change over time and the energy commodity demand. Furthermore, there may be restrictions on switches between modes. In particular, some mode switches may not be performed too frequently, which is modelled by a set of mode transition lockout timers (the state transition model has similarities with the timed automata formalism [7]). Note that many of these timers may not be relevant at the time scale that we consider (e.g. below a minute scale, which are relevant for more dynamic dispatch methods). In response to the control space (and a planning request), the energy application returns a schedule of mode switches for each actuator to EF-Pi.

To schedule *Buffers*, we use a model-based value iteration DP approach [51]. We only present the details that are specific to the problem at hand. Even though a buffer has only a single SoC variable, a *Buffer* has a large, multidimensional state space. To keep the DP tractable, approximations are needed to limit the size of the state space. Furthermore, the evaluation of the state space must be efficient.

To these ends, we precompute an internal simplified representation of the model. To limit the effort spent on preprocessing, we exploit that the system dynamics and mode transition model are time invariant. Thereby, we may reuse this representation between planning steps as long as the buffer system description (that is given by the device) not change.

The local optimization approach consists of the following steps. First, we translate an EFI buffer system description to an intermediate representation that is time invariant and time interval length invariant. The behaviour of a running mode influences the SoC, together with the running modes of the other actuators that are running at the same time and the losses. To simplify this, we determine the combined behaviour of every combination of running modes. We first make a product expansion of the possible state combinations in Section 5.4.3.1, followed by a combination of the fill level functions in Section 5.4.3.2. This gives a single fill level function for each combined state. As the fill level function depends on the SoC itself, the behaviour of this function may change within a time interval. To account for this, we determine the response of these functions for a fixed time interval length (described more technically as τ *projection*) or a set of time interval lengths in Section 5.4.3.3. This description is still time invariant and of a continuous form in the SoC dimension. Next, we discretize the response of the fill level function at specific SoC levels in Section 5.4.3.4 (using a fixed time interval length). This gives the time invariant model of the buffer system description that we use to efficiently solve the DP optimization problem. For the time intervals in the DP, there are also aspects that we can precompute without knowing e.g. the specific SoC or running mode, which we discuss in Section 5.4.3.5. This includes an evaluation of TRIANA's pricing scheme for each possible commodity consumption value. Based on this precomputed information, we can efficiently perform the backward induction of the Bellman equation of the DP in Section 5.4.3.6, which gives us a control policy that optimizes the device according to the pricing scheme and to the internal costs. To determine a planning for TRIANA and a control space allocation for EF-Pi, we perform a simulation of this policy in Section 5.4.3.7. Finally, Section 5.4.3.8 presents an extension to account for uncertain demand in this formulation. In the following, we discuss each of these steps in more detail.

5.4.3.1. AUTOMATA PRODUCT EXPANSION

Considering each of the actuators makes the DP state space expansion more complicated. Therefore, the first preprocessing step first simplifies the problem by expanding the Cartesian product of the actuator automata. Note that the product expands only the states, and not the combinations of transitions. The product expands only the states, and not the combinations of transitions, because the number of transition combinations per state grows exponentially with the number of parallel actuators. Figure 5.7 illustrates the expansion for a simple case.

For each actuator, a single timer lockout variable can represent the transition delays of the same length, and every transition blocks on all transition timers. To incorporate the leakage function, we translate it to an actuator with a single state.

FIGURE 5.7: Cartesian product of a 3-state automaton with a 2-state automaton, leading to a 6-state automaton.

FIGURE 5.8: Example of *Buffer* fill level function merging with two actuators and storage leakage. The functions map a fill level s (vertical) to a fill rate ds/dt (horizontal). The right-hand function represents the sum of the fill rate functions to the left.

5.4.3.2. FILL LEVEL FUNCTION MERGING

Each of the states in this Cartesian product represents a set of concurrent modes, one for each actuator. For each of these combinations, we can compute an aggregate fill level function that represents the joint behaviour. The DP can later use only this combined function, instead of using each of the actuator functions separately, which would be computationally expensive. To merge the piecewise constant functions, we use an event sort/merge algorithm. For each of the modes, the algorithm records the steps between the piecewise function segments.

Figure 5.8 illustrates the result of merging fill rate functions for a small example with two actuators, and a “leakage” pseudo-actuator to describe the loss over time. The example combines the *running* mode of a heat pump with the *on* mode of an

Input: set of fill level functions F
Output: merged fill level function f'

```

1 ·  $l_0, u \leftarrow \max\{\text{lb}(f) \mid f \in F\}, \min\{\text{ub}(f) \mid f \in F\}$  # lower, upper bound
2 ·  $r_0, c_0, \mathbf{x}_0 \leftarrow 0, 0, \mathbf{0}$  # fill rate  $ds/dt$ , cost rate  $dz/dt$ , demand per commodity
3 ·  $q \leftarrow \text{newPriorityQueue}()$  # ordered by lower bound
   # Record segment step events in  $q$ 
4 · for  $f \in F$ : # each fill level function
5 ·    $r, c, \mathbf{x} \leftarrow 0, 0, \mathbf{0}$ 
6 ·   for  $s \in f$ : # each segment
7 ·      $r_\Delta, c_\Delta, \mathbf{x}_\Delta \leftarrow s \rightarrow r - r, s \rightarrow c - c, s \rightarrow \mathbf{x} - \mathbf{x}$ 
8 ·     if  $s \rightarrow l < l_0$ : # before domain
9 ·        $r_0, c_0, \mathbf{x}_0 \leftarrow r_0 + r_\Delta, c_0 + c_\Delta, \mathbf{x}_0 + \mathbf{x}_\Delta$ 
10 ·    else:
11 ·       $q \leftarrow \langle s \rightarrow l, r_\Delta, c_\Delta, \mathbf{x}_\Delta \rangle$ 
12 ·     $r, c, \mathbf{x} \leftarrow r + r_\Delta, c + c_\Delta, \mathbf{x} + \mathbf{x}_\Delta$ 
13 ·  $q \leftarrow \langle u, \cdot, \cdot, \cdot \rangle$  # upper bound sentinel
   # Merge step events from  $q$ 
14 ·  $f' \leftarrow []$ 
15 ·  $s \rightarrow l, s \rightarrow r, s \rightarrow c, s \rightarrow \mathbf{x} \leftarrow l_0, r_0, c_0, \mathbf{x}_0$ 
16 ·  $s_\downarrow \leftarrow \square$ 
17 · for  $e \leftarrow q$ : # pop in order of  $e \rightarrow l$ 
18 ·   if  $e \rightarrow l > s \rightarrow l$ :
19 ·      $s \rightarrow u \leftarrow e \rightarrow l$ 
20 ·     if  $\max s \rightarrow \mathbf{x} - \mathbf{x}_{\max} \leq 0$ : # commodity consumption limit
21 ·       if  $s_\downarrow \neq \square$  and  $s_\downarrow \rightarrow r > 0$  and  $s \rightarrow r < 0$ : # convergence entry
22 ·          $s_0 \rightarrow l \leftarrow s \rightarrow l$ 
23 ·          $s_0 \rightarrow r, s_0 \rightarrow c, s_0 \rightarrow \mathbf{x} \leftarrow (s_\downarrow \rightarrow r \cdot \langle s \rightarrow r, s \rightarrow c, s \rightarrow \mathbf{x} \rangle -$ 
24 ·            $s \rightarrow r \cdot \langle s_\downarrow \rightarrow r, s_\downarrow \rightarrow c, s_\downarrow \rightarrow \mathbf{x} \rangle) \cdot (s_\downarrow \rightarrow r - s \rightarrow r)^{-1}$ 
25 ·          $f' \leftarrow f' + [s_0]$ 
26 ·          $f' \leftarrow f' + [s]$  # add segment  $s$ 
27 ·          $s_\downarrow \leftarrow s$ 
28 ·       else:
29 ·          $s_\downarrow \leftarrow \square$ 
30 ·        $s \rightarrow l \leftarrow e \rightarrow l$ 
31 ·       if  $e \rightarrow l \geq u$ :
32 ·         break
33 ·    $s \rightarrow r, s \rightarrow c, s \rightarrow \mathbf{x} \leftarrow s \rightarrow r + e \rightarrow r, s \rightarrow c + e \rightarrow c, s \rightarrow \mathbf{x} + e \rightarrow \mathbf{x}$ 
34 · return  $f'$ 

```

ALGORITHM 5.1: *Buffer CSA fill level function merging.* Note that the result is independent of a specific time interval.

FIGURE 5.9: Example of *Buffer CSA* τ forward projection (lower part) and backward projection (upper part), for an exaggerated value of τ . In both presented cases, the projection from the a side “wins”.

auxiliary heating rod (plus storage leakage over time). The figure gives the fill rate (ds/dt , horizontal) as a function of the fill level (s , vertical). In a similar manner, the figures for the functions of s to commodity demand, and s to internal cost can be constructed (note that the pieces of these functions are structurally linked). The illustrated heating rod has a wider operating range than the heat pump. However, in a combination of modes, the domain intersection determines the domain of the composite function. Furthermore, the *Buffer* control space can impose limits on the combined demand, leaving gaps in the domain (not shown in this example).

Algorithm 5.1 describes the procedure more technically. The algorithm represents the change at each of the break points of the functions, and then applies these changes in the right order. The ordering by fill level is implemented with a heap. Special care is needed to account for the lower and upper bound values. At the boundary between an increasing and a decreasing segment, we insert a point to represent the oscillation around that fill level (in the following, we now know that this transition no longer occurs). The time complexity of this algorithm is determined by the cost of the priority queue operation (implemented by a heap), which is in $\mathcal{O}(\log n)$. For a problem with n_s pieces in the modes at hand, the time complexity of the complete algorithm is in $\mathcal{O}(n_s \log n_s)$ (with minor changes, the algorithm also applies to piecewise linear functions, and to convex combinations of functions). After this operation, we delete infeasible modes, i.e. modes with fill level functions that have no pieces.

5.4.3.3. PROJECTION

Over time, the system can traverse through multiple pieces of the piecewise constant function, which can be inefficient to evaluate. Due to the structure of these functions, for a given time interval length τ , we can derive a system response model that is piecewise linear in the fill level at the begin of the time interval.

Algorithm 5.2 presents a procedure to determine this piecewise function in $\mathcal{O}(n_{s'})$, where $n_{s'}$ is the number of pieces in the product automaton response function. The projection steps of the algorithm are also illustrated in Figure 5.9. The algorithm exploits the property that the rate of change in s of each of the parameters only depends on the parameters of the “left-hand” ($t = 0$) side (labelled a) and the “right-hand” ($t = \tau$) side (labelled b) piece; all pieces in between give only a constant offset. Intuitively, the update step shifts the block between the segments as far as possible to the left, while recording the change to the parameters at the left-hand side and the right-hand side. The shift stops when s_a or s_b reaches the upper bound of respectively either the a or the b segment (the algorithm uses a less intuitive closed form). In every step, at least one of the values of p_a and p_b increases. Due to this monotonic progress, the algorithm needs at most $2n_{s'}$ steps to visit all pieces, which gives the algorithm a linear complexity. Whereas the algorithm is trivial in time complexity, the handling of all combinations of fill rate transitions (positive to negative, negative to zero, etc.) is cumbersome. The concept of the algorithm applies to any piecewise monotonous function, as long as the algorithm can find the upper bound intercept points and an expression of the relation between the change in s and the value change at τ for each of the parameters. The class of piecewise linear functions may be of particular interest for future work (if ds/dt is linear in s , then s is exponential in t ; note that break points should be inserted at sign changes to ensure monotonicity of the pieces).

Although the system response also depends on the given demand for the buffer, we choose not to include this demand in the model, because it results in non-(piecewise)-linear equations. Instead, we approximate the buffer demand effect in the DP. The approximation may be improved by computing the response for a set of reference buffer demand levels, and using the closest of these references for each DP time interval.

5.4.3.4. DISCRETIZED RESPONSE

Next to the continuous system response, we also precompute the discrete system response, which we use in the DP. For every mode, we discretize the fill levels, and compute the specific parameters (i.e. apply f_τ) of these fill levels. Each discrete state point represents the response in a part of the state space, whereby states do not have to align exactly to these representative states. Furthermore, for each state we need to be able to approximate the DP successor state. Rather than adding a direct edge to the next state point, since we want to be able to account for demand, we use real (fixed-point) numbers to represent state differences. The actual successor state point depends on the SoC at the start of the time interval and the demand level.

At this point, the discretized model represents only the possible immediate state changes, independent of the more dynamic aspects of the system over time, i.e. the actual buffer demand, the commodity demand cost and the possible values of lockout timers. These aspects are integrated in the problem during the state space expansion step.

Input: merged fill level function f , time interval length τ

Output: τ -projected fill level function f_τ

```

1 ·  $f_\tau \leftarrow []$ 
2 ·  $p_a \leftarrow p_b \leftarrow \text{lowest}(f)$  # left-hand, right-hand piece
3 ·  $s_a, s_b \leftarrow p_a \rightarrow l, p_b \rightarrow l$  # left-hand, right-hand SoC
4 · while  $p_a \neq \square$ :
    # Initial projection
    5 ·  $d \leftarrow \text{reference to } \begin{cases} b, & \text{if } p_a \rightarrow r \geq 0 \\ a, & \text{if } p_a \rightarrow r < 0 \end{cases}$  # projection direction
    6 ·  $r, c, \mathbf{x} = 0, 0, \mathbf{0}$ 
    7 ·  $t \leftarrow \tau$ 
    8 · while  $t > 0$  and  $p_d \neq \square$  and  $s_d \geq p_d \rightarrow l$ :
        9 ·  $t_d \leftarrow \min\{t, \begin{cases} (p_d \rightarrow u - s_d) / |p_d \rightarrow r|, & \text{if } p_d \rightarrow r \neq 0 \\ \infty, & \text{if } p_d \rightarrow r = 0 \end{cases}\}$ 
        10 ·  $r, c, \mathbf{x} \leftarrow \langle r, c, \mathbf{x} \rangle + \langle p_d \rightarrow r, p_d \rightarrow c, p_d \rightarrow \mathbf{x} \rangle \cdot t_d$  # by element
        11 ·  $s_d \leftarrow s_d + |p_d \rightarrow r| \cdot t_d$ 
        12 ·  $t \leftarrow t - t_d$ 
        13 · if  $s_d = p_d \rightarrow u$  and  $p_d \rightarrow r \neq 0$ :
            14 ·  $p_d \leftarrow \text{next}(p_d)$ 
        15 · if  $t > 0$ : # projection infeasible
            16 ·  $p_a \leftarrow p_b \leftarrow p_d$ 
            17 ·  $s_a \leftarrow s_b \leftarrow p_d \rightarrow l$ 
            18 · continue
        # Update projection until next discontinuity
        19 · while  $p_a \neq \square$  and  $p_b \neq \square$  and  $s_a \geq p_a \rightarrow l$  and  $s_b \geq p_b \rightarrow l$ :
            20 · if  $p_a \rightarrow r = 0$  and  $p_b \rightarrow r < 0$ : # inversion swap
                21 ·  $p_a, p_b \leftarrow p_b, p_a$ 
            22 ·  $r', c', \mathbf{x}' \leftarrow \langle p_b \rightarrow r - p_a \rightarrow r, p_b \rightarrow c - p_a \rightarrow c, p_b \rightarrow \mathbf{x} - p_a \rightarrow \mathbf{x} \rangle \cdot \begin{cases} p_a \rightarrow r^{-1}, & \text{if } p_a \rightarrow r \neq 0 \\ \langle 0, 0, \mathbf{0} \rangle, & \text{if } p_a \rightarrow r = 0 \end{cases}$ 
            23 ·  $h_a \leftarrow \min\{p_a \rightarrow u - s_a, \begin{cases} (p_b \rightarrow u - s_b) \cdot p_a \rightarrow r / p_b \rightarrow r, & \text{if } p_b \rightarrow r \neq 0 \\ \infty, & \text{if } p_b \rightarrow r = 0 \end{cases}\}$ 
            24 · if  $h_a < 0$ :
                25 · break
            26 ·  $f_\tau \leftarrow f_\tau \cup [\langle s_a, s_a + h_a, r, r', c, c', \mathbf{x} / \tau, \mathbf{x}' / \tau \rangle]$ 
            27 ·  $s_a, s_b \leftarrow s_a + h_a, s_b + h_a(1 + r')$ 
            28 ·  $r, c, \mathbf{x} \leftarrow r + h_a r', c + h_a c', \mathbf{x} + h_a \mathbf{x}'$ 
            29 · if  $s_a = p_a \rightarrow u$ :
                30 ·  $p_a = \text{next}(p_a)$ 
            31 · if  $s_b = p_b \rightarrow u$ :
                32 ·  $p_b = \text{next}(p_b)$ 
        # Restart
        33 ·  $p_a, s_a \leftarrow p_b, s_b$  if  $s_b > s_a$  or  $p_b > p_a$  else  $p_b, s_b \leftarrow p_a, s_a$ 

```

ALGORITHM 5.2: Buffer CSA τ projection.

5.4.3.5. STATE INVARIANT PREPROCESSING

Before the state space expansion, we precompute the time (but not state) variant parameters of the problem. Evaluation of commodity demand costs for each automaton state, for a duration of τ .

The buffer demand is approximated as a step change at the end of a time interval. The step change equals the aggregate demand in the (continuous) half time interval before and after this point in time. We could reuse the demand vector for multiple planning iterations with the same time offset, but the recomputation overhead is negligible.

We can determine the commodity consumption cost in advance for every possible commodity consumption vector, based on a list of possible commodity consumption values that have been determined within the earlier discretized response step. Due to the fill level function projection, discrete state points may often have different consumption vectors, depending on the fill rate pattern in the fill level function. The vector will still be shared between different lockout timer states. In the DP evaluation, we can now determine the cost of a control option with a LUT lookup.

5.4.3.6. BACKWARD STATE SPACE EXPANSION

After the preprocessing, we can explore the state space with reasonable efficiency. We use a value iteration approach. The state space has a structure with internal edges within a DP phase, which means that we have to evaluate a phase iteratively until there are no edge changes.

We represent states with a tuple $\langle t, m, s, [\tau_1, \dots, \tau_{n_\tau}] \rangle$, where $t \in \{1, \dots, n_t\}$ is the time interval, m is the mode, s is the discretized fill level and τ_i is the lockout timer state for $i \in \{1, \dots, n_\tau\}$, where n_τ is the number of mode transition lockout timers. Note that lockout timer support is until now only actually implemented for the *Unconstrained* class, which is derived from the *Buffer* class. The DP phases correspond to time intervals, which means that we can keep the time interval t implicit in the state description. The mode m gives the mode “just before” time interval t .

The DP state space expansion starts from the back at $t = n_t$, and iteratively gives an approximation of the cost function at $t - 1$ until we have an approximation of the cost function at $t = 0$, in accordance to the Bellman equation. We use a continuous interpretation of the fill level space, and use linear interpolation to determine the cost between the explicitly considered points. The end state cost function penalizes quadratic distance from the start fill level to limit the horizon effect. We do not use a reachability analysis (i.e. forward sweep). The expansion of a time interval consists of two sequential phases: a transition phase and a passage-of-time (τ) phase. These phases are evaluated in reverse order. The transition phase evaluates all transition options at a specific point in time, using an iterative approach to determine the optimal sequence of transitions for each mode, from each point. We consider the transitions of each actuator separately to avoid a combinatorial increase in the number of possible actions. The τ phase applies the effect of staying in a mode from a point, and uses the cost function at the next time interval.

At this point, the state space describes the (approximation of the) optimal policy under the given pricing regime, which is used in the policy simulation step.

5.4.3.7. POLICY SIMULATION

The policy gives a mapping from individual state descriptions to actions, which gives sufficient information to execute the plan. However, for coordination TRIANA needs an estimate of the commodity demand over time. To make this estimation, we use the model to simulate the execution of the policy over time. In this (deterministic) simulation, we sequentially round the current state to the closest state point, apply the control action that has been determined in the backward sweep, and repeat this process until we reach the end of the optimization horizon. The simulation produces the EF-Pi switch actions for the allocation as well.

The simulation may reach a state for which no policy is available, i.e. that was not covered by the state space exploration. In this case, the missed state may be added to the state space, and the exploration can be repeated. If the missed state is infeasible, then the state may be added with a large penalty, or the sequence leading up to the infeasibility may be discounted. The current implementation does not implement these repair strategies, and instead decides to omit the planning and the allocation for TRIANA and EF-Pi, respectively (if no planning is given, the behaviour of the node should be considered to be uncertain; omitting the allocation should result in a default fail safe execution).

The execution trace may be used as feedback to refine the policy. For example, the search can be repeated with a smaller τ in a restricted space, or the trace can bootstrap a local search algorithm. We have not implemented policy refinement techniques.

The planner returns the resulting commodity demand pattern of the simulation to the TRIANA client callback. We give the allocation to EF-Pi only after the parent TRIANA aggregator, or its local substitute, commits to the pricing scheme/pattern pair.

The complete policy map is large; simultaneously storing the map for every device limits the number of devices that a single machine can host (the implementation uses a semaphore to protect the use of the memory; at the moment, this semaphore mainly serves to reduce cache and VM thrashing). To avoid storing the policy map, we discard the policy after the simulation, and recompute it when needed (we cache the pricing scheme block and the EF-Pi allocation). More elaborate space/time tradeoffs may be considered in the future, for example by keeping the policy for the first few time intervals or using a more compact (e.g. compressed) policy representation.

5.4.3.8. UNCERTAIN DEMAND

The heat demand of a buffer is uncertain. Therefore, the buffer usage forecast specifies the confidence interval on the prediction of the heat demand, expressed as a normal distribution value for the energy demand in a time interval. Although

we may in principle consider a stochastic DP to address this uncertainty, we use a more simple method to limit the computational cost of supporting these forecasts with uncertainty. We derive a simplified model to limit the risk associated with the uncertainty of demand. The idea is that we “mirror” the uncertainty on the upper and lower bound of the buffer. We reason about the expected feasible range, and add a penalty for residing in parts of the state space that have a significant chance of becoming infeasible. This approach assumes that the feasible range is fairly smooth, i.e. the behaviour and costs of using a device should not change suddenly when the SoC changes. This penalty can be represented by a single vector per time interval (one point per n_b), and is inexpensive to compute and to apply.

EF-Pi allows a driver to specify that the SoC has to be within a specific range at certain times, i.e. a time variant upper and lower SoC bound. We may add support for these target SoC profiles with the same mechanism (e.g. the SoC of an EV in the morning, or a desired temperature range). However, to combine both, we should take the convolution of the uncertain range with the target SoC costs to give an appropriate representation of both aspects.

5.4.4. UNCONSTRAINED CSA

The *Unconstrained* control space represents a subset of the *Buffer* control space class. As *Unconstrained* control spaces do not have storage and therefore also have no buffer demand, the scheduling can be simplified considerably. An *Unconstrained* control space may be seen as a *Buffer* with no SoC variable and a single actuator.

In contrast to the *Buffer* control space, we have implemented support for the lockout timers for the *Unconstrained* class. It should be straightforward to port this approach to the *Buffer* class.

5.4.4.1. LOCKOUT TIMERS

Mode transition lockout timers can, in the worst case, lead to a combinatorial growth of the state space. Therefore, a suitable compact representation of the timers is important. We abstract timers to a *discrete time Markov chain* (DTMC) model (based on a *continuous time Markov chain* (CTMC) model): rather than representing the explicit timer countdown, we model the timer expiration as a memoryless random process with an average timeout of the timer duration. If a timer expires in a duration of less than a single time interval, then we may simply ignore it. After each time interval, there is a probability of τ/τ_i that a timer i expires, for $i \in \{1, \dots, n_T\}$. The probability is represented with weights in the DP evaluation. The DTMC abstraction reduces the state space of each timer to two states, i.e. *active* and *inactive*. A possible more accurate alternative may model several points in the time countdown. However, this model increases the number of states exponentially, and we see no justification for this accuracy improvement at this moment.

In our DP representation, each timer is represented as a binary dimension that expires according to an exponential distribution. Consequently, the state space grows by a factor 2^n with the number of timers. We observe that the timer expira-

tion problem is separable: for each timer, we can separately consider whether this particular timer expires, and compute the expected costs for this. Hereby, we can compute the expected cost in n steps, each of which has to consider $\mathcal{O}(2^n)$ states.

This approach still leads to an exponential growth of states in the number of timers. However, in practice we can reduce the number of timers by considering only the timers that are actually violated in the proposed optimal solution, which reduces the value of n . Adding a timer can only result in a worse solution, because it adds a constraint to the optimization problem. First, we solve the problem without considering any of the timers. If the proposed solution actually violates any of the timers, then we put this timer in the active timer set, add a dimension to the problem (increasing n by 1), and solve the problem again. Usually, the problem is solved at $n = 1$ or $n = 2$, depending on the structure of the problem.

5.5. EXPERIMENTS

The performance of the approach presented in the previous section is sensitive, both in terms of planning performance (the quality of the outcome) and the computational cost (execution time and memory use). We use simulations to quantitatively evaluate (our implementation of) TRIANA-EF-Pi.

In this section, we focus on the implementation of the device DPs. In particular, we choose to look at the performance of the implementation of *Buffer* control space planning, because it dominates all other relevant cases in terms of execution time. The *Unconstrained* control space behaves similar in extreme cases, but avoids the fill level dimension. The execution time of the other DPs is typically negligible, and thereby not of immediate interest. We omit measurements on the reimplementations of the TRIANA core functionalities, since these perform similar to the original implementations in terms of planning quality and do not introduce new computational challenges.

5.5.1. SCHEDULING PERFORMANCE

5.5.1.1. EXPERIMENT

The *Buffer* CSA DP has five parameters: the control space, the initial state, the pricing scheme, the time discretization step τ , and the number of points n_b that are used to discretize the fill level. To experiment with the behaviour of the DP implementation, we set up a simple experiment. We use the control space of a battery with variable efficiency, which is initially half-full and idle. We apply a sinusoidal linear price signal with three periods on a day ahead horizon ($c(t) = 2 + \sin(-3t \frac{2\pi}{24 \cdot 60 \cdot 60})$, t in seconds, $0 \leq t < 24 \cdot 60 \cdot 60$).

In the experiment, we take τ and n_b as variables. Note that the appropriate values for τ and n_b may change depending on the structure of the control space and the pricing scheme. For example, τ needs to account for both the dynamics in the pricing scheme and the model. The fill level point count n_b should be high enough to accurately represent the cost function and resulting policy.

To give an indication of the quality of the planning, we use an LP model to give a lower bound on the optimal objective value. The LP model does not account for the variable efficiency, and instead assumes the highest efficiency for the full range.

5.5.1.2. RESULTS

Figure 5.10 presents the results of the simulation experiments. The solution quality is expressed by the objective value of the optimization problem. These measurements confirm that the time interval length and the number of different fill levels should be chosen high enough to represent the dynamics of the model. At some point, decreasing τ and increasing n_b do no longer meaningfully change the decisions. With a decrease in time interval length, the objective value appears to asymptotically approach a value, which is presumably the (true) optimal solution. This value is close to the LP optimal value bound (1.5% higher).

An increase of n_b does not always lead to better results, especially for smaller values of n_b . The choice of the exact location of the points affects the approximation, which can in some cases give worse results. The choice of τ can affect the approximation similarly; however, in this case regressions of the objective value are rare. As a tradeoff between computation time and resource use, we typically choose $\tau = 900$ s (15 minutes) and $n_b = 30$.

5.5.2. PLANNING RESOURCE USE

5.5.2.1. EXPERIMENT

To measure the resource use of the DP, we provide the algorithm with increasingly complex control space instances. We generate these instances by connecting copies of a 3-state actuator to the same *Buffer*. We denote the number of actuators by n_a . Figure 5.11 illustrates the automaton and its expansion for $n_a = 2$.

To measure the memory use, we take the memory use as reported by the Java runtime after a forced garbage collection. Because the time overhead of this forced collection is substantial (i.e. milliseconds), we start the memory measurements from $n_a = 6$. We also provide a lower bound on memory use by analyzing the data structures. Each automaton state gives $n_t \cdot n_b$ DP states, each of which takes up 12 bytes in arrays on a 64 bit system. For the experiment, this means that each automaton state costs at least 35 kB in DP state space memory. The number of states in the product automaton is 3^{n_a} .

The DP state space size can still be reduced significantly, to 4 bytes per state, if we would not store the actual decision. Discarding the decision is in principle not a problem, because the decision can be recomputed efficiently during the forward sweep, using the minimum cost information that is still available. However, to avoid code duplication, we choose not to do this. An alternative approach that we recommend as a first alternative if the memory use becomes a problem is to store the indices instead of references to the decision modes: a byte array can record up to 256 control options, which should be sufficient in most practical cases (but – with a direct encoding – not for the benchmarks at hand), and leads to 5 bytes of storage per DP state.

The experiments are performed on a 2011 laptop PC running Linux with an Intel Core i5 M540 (2 cores, 4 threads) and 4 GB of RAM. For a single planning problem, the implementation uses a single CPU thread.

FIGURE 5.10: Objective value (lower is better) as a function of n_b and τ .

FIGURE 5.11: Automaton of Section 5.5.2 with its Cartesian product expansion for $n_a = 2$.

FIGURE 5.12: Computational load (left axis) and memory use (right axis) as a function of n_a combined actuators ($n_t = 96$, $n_b = 30$). Note that the scale on both vertical axes is logarithmic.

5.5.2.2. RESULTS

Figure 5.12 presents the measurements of run time and the memory use for different numbers of parallel actuators. The exponential growth pattern is evident. There is a small fixed run time overhead in the planning and logging that is visible at $n_a = 1$, but after that the DP expansion dominates. The run time grows with approximately a factor 5 per actuator. In contrast, memory use grows with a factor 3 per actuator. The measured memory use values correspond well to the estimates given in advance.

There are no serious jumps in run time, which suggests that the growth in memory use does not result in significant thrashing. An explanation for this is that the DP uses relatively small segments of memory at the same time, which makes caching effective. There is a slight jump from $n_a = 7$ to $n_a = 8$ (factor 7), which we believe to come from excessive garbage collection. We expect disk thrashing if the problem is so large that pairs of DP phases (time intervals, i.e., the current and the successor states) do not fit in RAM. The PC runs out of memory at $n_a = 11$.

5.5.2.3. EVALUATION

For the control spaces we anticipate for practical use, the presented DP formulation has an acceptable cost in run time and memory use. The extensive preprocessing helps to make the DP expansion efficient. The planning can be considered to operate in reasonable time for interactive control from the perspective of TRIANA up to about $n_a = 6$. The device control space planning should be fast, because TRIANA's group planning uses the device planning several times, such that it can choose from multiple possible plans for the device. For larger instances, the planning is still useful to find a better "baseload" pattern for the devices with a more manageable structure, until memory is exhausted.

The experiments show that in the used formulation, estimating the memory use is easy, and run time can be estimated as well. Depending on the circumstances, a controller can make tradeoffs in the parameters of the planning, i.e. it can choose between a fast and an accurate planning. A "smart" controller that manages the use of the planning algorithms is left as future work.

While the experiment only considers concurrent automata as the source of exponential growth in computation time and memory use, timers give an exponential increase in both as well (with a factor 2). Therefore, control spaces with many timers are intractable for this DP formulation. To support such control spaces, a future version may use more coarse abstractions, for example by combining similar timers, at the cost of accuracy.

5.5.3. MULTICOMMODITY OPTIMIZATION CASE

5.5.3.1. SCENARIO

To evaluate the ability to profit from the availability of multiple commodities, we consider a scenario with a single hybrid heat pump that can choose to supply heat either with a heat pump or with a gas heater, simulating a physical set-up at the Energy

FIGURE 5.13: Hybrid heat pump configuration (modelled as a single *Buffer* device) with separate actuators.

Transition Centre in Groningen, the Netherlands. The physical experiments on this specific platform have not yet been performed due to driver development issues. Note that we have also performed some preliminary experiments with the balancing of both the electricity production and the gas demand of a large group of microCHPs and heat pumps, although the results of these experiments are not presented in this thesis.

We set up the simulations as follows. First, we configure a fixed price profile. Subsequently, we set up the EF-Pi configuration as presented in Figure 5.13. Next, we simulate the system for one week. Finally, we record the measured cost. Our baseline case is a conventional, set point driven control. Since we do need to record the control space messages to be able to record the costs, we use a modified version of TRIANA that does not send allocations. In return, devices respond as if conventional, set point driven control should be performed. We make separate baseline simulations for a case with only a gas heater and a heat pump, as we do not want to define a specific control strategy to choose between these two devices. Furthermore, in the optimized case, we also simulate the system without a gas heater. By this, we show that the hybrid aspect contributes to the efficient use of the system.

As a price reference, we use the APX-NL spot market clearing prices. We used the dataset from the year 2011, as this dataset was available in a convenient format. In more recent data, prices are more volatile, which is beneficial for optimization. To account for the price difference between spot prices and retail prices, we normalize the prices to the retail price. Arguably, a static offset should be added before normalization to account for energy taxes. While we assume that the APX prices reflect the prices that we have to pay for the consumption of electricity, the prices are actually the outcome of a day-ahead auction. To exploit these prices in reality, it would be necessary to participate in the auction, or to find a retailer who is willing and able to follow this market outcome.

While the COP for a heat pump is temperature dependent in a real heat pump, due to limitations in the used buffer simulation component we use a fixed efficiency for the complete range. A variable efficiency is supported by EF-Pi, but not configurable in the used buffer simulation component. The parameters for the simulation are as follows:

- Simulation control
 - Duration: 1 week (starting from 2012-01-01 10:00)
- Pricing scheme
 - Electricity:
 - * Dynamic: APX-NL pricing scheme, dataset from 2011-01-01 0:00 to 2011-31-12 23:59, hourly prices, shifted by 1 year, normalized to €0.22/kWh mean in first week
 - * Static: €0.22/kWh
 - Gas: static, €0.614/m³
- Aggregator
 - Sinusoid base load: 1000 W (2000 W peak-to-peak)
- Buffer
 - Dimension: 10 kWh ($\approx 200 \ell$)
 - Initial state of charge: 5 kWh
- Gas heater
 - Heat production: 15 kW
 - Coefficient of performance: 0.9 (static)
 - On transition cost: €0.025
 - Off transition cost: €0.025
- Heat pump
 - Electricity consumption: 1000 W
 - Coefficient of performance: 3.0 (static)
 - On transition cost: €0.05
 - Off transition cost: €0.05
- Demand
 - Static load: 1500 W⁴

4. The demand has been simulated by a leakage function due to limitations in the *Buffer* component that was used for the simulations.

	1. Gas only	2. HP only	3. Opt.	4. Opt. APX	5. Opt. APX, no gas
electricity	€0.00	€17.92	€13.25	€10.45	€14.51
gas	€18.90	€0.00	€5.16	€3.34	€0.00
switching	€1.15	€1.25	€1.65	€2.00	€2.10
total	€20.05	€19.17	€20.07	€15.79	€16.61
	+4.6 %	(+0 %)	+4.7 %	-17.6 %	-13.4 %

TABLE 5.1: Simulation results of multicommodity optimization case, by scenario (horizontal), broken down by cost source (vertical).

The scenario configurations for the simulations are as follows:

1. Baseline gas heater
No optimization, completely fill buffer with gas heater when buffer becomes empty. Only use static gas price to determine costs.
2. Baseline heat pump
No optimization, completely fill buffer with heat pump when buffer becomes empty. Only use APX prices to determine costs.
3. TRIANA optimization, static electricity prices
Buffer optimization with TRIANA energy application. We use the average electricity price to optimize against, but use APX prices to determine actual costs.
4. TRIANA optimization, dynamic electricity prices
Buffer optimization with TRIANA energy application. We optimize against APX electricity prices, and use the same prices to determine actual costs.
5. TRIANA optimization, dynamic electricity prices, only heat pump
Buffer optimization with TRIANA energy application. We optimize against APX electricity prices, and use the same prices to determine actual costs. To confirm that the gas heater contributes to the result of the previous scenario, we leave out the gas heater in this scenario.

The first two scenarios serve as baseline scenarios. The third scenario serves for validation. The fifth scenario validates that the hybrid aspect is exploited. The simulation has a duration of one week. We present the accumulated cost, broken down by cost source. All costs are in euros, for one household, for a period of one week in winter.

5.5.3.2. RESULTS

The results of the week simulation are presented in Table 5.1. We first consider the baseline scenarios, which use either only the gas based heating or the electric heat pump based heating. The small difference in costs between the baseload (gas-only

and electricity-only without control) scenarios show that the gas heater based and the heat pump based heating solution are mutually competitive in terms of costs. Whereas the switching costs for the heat pump are twice as high as for the gas heater, the switching costs are almost equal. The explanation for this is that the heat pump produces heat far more slowly, allowing the heat pump to run for a much longer period than the gas heater.

The “optimized” control of TRIANA using a static electricity price for optimization is actually slightly more expensive than the baseline scenarios. We have several possible explanations. First, due to discretization inaccuracies, TRIANA can not exploit the entire range of the buffer. This means that it can not completely exploit low price periods (which is not relevant in this scenario), and that it needs to switch more often to keep the buffer within range. The same inaccuracies may also account for the use of the gas heater, while use of only the heat pump would have been more cost effective. Finally, since the simulation period is fairly short, it may inadvertently have switched on at times with high electricity prices (i.e. “bad luck” in the timing of the switch actions in relation to the pricing). More extensive simulations can exclude this possible cause.

The optimized control with TRIANA, using the actual market prices, shows the expected substantial decrease in cost. As only the electricity prices (and not the gas prices) are dynamic in the scenario, the control exploits periods with low electricity prices, which shifts the system from the use of gas to the use of the heat pump instead of the gas heating. The gas heating is only used in periods where the buffer is not large enough to exploit low prices, serving as a back-up source. To exploit the low prices, the heat pump has to switch far more often.

The fifth scenario considers a case where we can perform price optimization but may not use the gas heater, which shows that the hybrid aspect actually lowers cost, i.e. if the gas heater is not available, then costs increase. However, in the current market situation, periods with extreme prices are rare. Therefore, the benefit of the ability to revert to gas heating is limited, although still substantial (4.2 percentage points).

Some caveats should be noted regarding the simulated scenario. The simplified buffer simulation does not account for the variable efficiency of the buffer (even though both EF-Pi and our *Buffer* implementation support this). Hereby, especially scenario 2 and scenario 5 are too optimistic about the capabilities of the buffer. Also, in practice, a significant part of the capacity should be reserved to cope with uncertainty in demand, as the heat pump is not able to supply all heat as demand occurs. In contrast, if a gas heater is available, then this heater can respond to high demand, and thereby increase the usable buffer capacity. Consequently, we expect that the profit increases on scenarios that correspond more closely with reality.

5.6. CONCLUSION

The implementation of TRIANA-EF-Pi demonstrates that TRIANA concept adapts well to an environment that targets a practical setting. Furthermore, it shows that EF-Pi can host DSM approaches other than the one for which it was originally

developed. This confirms the belief of the developers of EF-Pi that this platform offers a flexible, general implementation environment for demand side management. Our work significantly increases the credibility of EF-Pi as a platform that can support a wide variety of energy management applications.

In addition to the more theoretical challenges that have not been resolved in Chapter 3, there are quite some loose ends in the implementation that may affect its practical usefulness. For example, the current implementation of *Time Shiftables* in TRIANA-EF-Pi supports only single segments (although we have described how multi-segment support can be (re)introduced), and does not account for already running devices. The timers have been implemented only for the *Unconstrained* class, and not for the *Buffer* class. Since the structure of these classes is very similar (and the same in relation to the timers), porting this timer support should in principle be straightforward.

The versatility of EF-Pi has allowed us to apply TRIANA-EF-Pi in three different research projects: HEGRID [188], VIOS [320] and e-balance [87]. In these projects, we have run TRIANA-EF-Pi on low cost embedded Linux platforms. This reusability significantly reduces the engineering effort associated with practical experiments, and thereby allows us to focus on the problem at hand.

5.6.1. RECOMMENDATIONS

5.6.1.1. IMPLEMENTATION ROBUSTNESS

The robustness of the planning mechanism (i.e. the computation and communication process) should be improved for a real-world setting: the planning is restartable (and is restarted every 5 minutes), but it does not yet recognize stalling, e.g. due to a single non-responsive child. A practical implementation should retry and/or ignore these children. There is no event-driven local planning yet (e.g. in response to high PV feed-in): most device drivers send an update e.g. every 10 seconds, which means that we have to use some policy to decide when to respond to an event. The planning process is currently driven by the root node of the planning tree. The design does account for support for such an approach, and a local replanning may be triggered manually in the current version. Also, there is no checking on the values that are communicated within the planning process, which means that a single misbehaving child (either due to an implementation fault or out of malice) can corrupt the planning process. Security is virtually nonexistent, and relies on the protection of the boundaries of the system. While these problems are acceptable for small-scale field test applications in a controlled environment, TRIANA-EF-Pi is far from production ready at this point.

5.6.1.2. INFRASTRUCTURE SPECIFICATION

EFI focuses on the energy flexibility within a household, and assumes that all demand is connected at the same point. In larger buildings and at higher levels in the energy infrastructure, the organization of the energy structure has a relevant effect as well. Note that this in some cases already plays a role at a household level:

for example, some jurisdictions do not allow feed-in to the grid with a battery. Also, it is not always completely clear where a smart meter is connected (i.e. for which devices the demand is measured). It may be useful to have a common description language for describing the configuration of energy systems, in addition to the energy flexibility descriptions of the resources themselves. Also, a common language for the objectives and constraints of energy systems (with more structure than just a mathematical description) would be useful to increase the interchangeability of control systems.

5.6.1.3. DEVICE LEVEL FORECASTING

Currently, the responsibility to provide device level energy flexibility forecasts is placed in the device drivers. However, in practice driver developers do not appear to be able to provide useful forecasts, and do not have a feel for how this forecasting affects energy applications (the device “works”, so the driver is finished). This reinforces the idea that device drivers should be treated more as a way to deal with the implementation complexities of specific devices, rather than as intelligent agents that represent a device. The intelligence should be either in the energy application or in a common library. A place where the “forecast” does work well is the communication of user-specified bounds, e.g. a charging or washing machine deadline, although this forecast becomes available only at the time that it is specified.

5.6.1.4. SIMULATION AND BENCHMARKING

EF-Pi is designed as a practical platform for DSM, and offers limited facilities for the simulation and benchmarking of DSM implementations. For validation as well as research purposes, a more complete simulation environment is desirable. We think that EF-Pi may serve as a suitable host for this.

An implementation of Flex Street on EF-Pi gives an interesting opportunity for a fair comparison between DSM approaches, without extra effort on behalf of the DSM application developers. This enables a competitive environment where it is possible to easily compare DSM approaches and to track their progress.

CONCLUSION

summary 229 · conclusions 232 · contributions 234 · recommendations 235

DEMAND SIDE MANAGEMENT BASED FLEXIBILITY IN SMART GRIDS brings huge benefits to the energy world, yet also brings huge challenges. This thesis lays a foundation for the further development of a flexible, effective and efficient coordination approach for flexibility in smart grids. The use of improved planning techniques and addressing various practical problems in the application of optimization based DSM control brings the dream of DSM – and thereby the cost effective implementation of the energy transition – a bit closer to reality.

Although in this thesis we have made some progress towards effective, efficient and realizable coordination of flexibility in smart grids, we are still far away from the universal smart energy coordination solution that we strive for, and it may take another decade of research before this problem is satisfactorily solved in general. In the meantime, advances in computation, communication and energy hardware, and picking the low hanging fruit first (i.e. consider very flexible devices and easy, forgiving applications), may simplify the problem, such that DSM may already start to contribute to the energy transition.

In the following, in Section 6.1 we first give an overview of the work in this thesis, followed by the answers to the original research questions in Section 6.2. We outline the contributions in Section 6.3. Finally, we give recommendations for future work in Section 6.4, in addition to the more specific recommendations already given in the previous chapters.

6.1. SUMMARY

6.1.1. COORDINATION IN SMART GRIDS

Chapter 2

The energy domain is a large and complex environment. The conventional grid extensively relies on mathematical optimization for the planning and control of the available resources, which has become less direct since the deregulation of the electricity market. Power system optimization is a mature field in comparison to control algorithms for smart grids, which is far from settled. Smart grid control tends to use more dynamic, just-in-time control methods, which are generally hard to optimize.

TRIANA is a realization of a planning based smart grid control concept that uses concepts similar to those used in conventional power system optimization. TRIANA partitions the optimization problem according to the hierarchical structure of the electricity grid, and splits up the DSM control problem in three phases: forecasting, planning, and real-time control. Although the approach is scalable and conceptually elegant, it simplifies the problem to such an extent that the solutions are sometimes far from optimal.

6.1.2. COORDINATION IN TIME

Chapter 3

Continuing the work on TRIANA, we study the hierarchical decentralized optimization of large groups of devices based on the aggregation of demand profiles, similar to methods that were used 30 years ago for the planning of the conventional grid, which can nowadays easily be solved on low cost embedded systems. However, in smart grids the scales and device properties are quite different, and therefore existing methods can not be used directly. We set up a common model for these optimization algorithms, and present three algorithms with different characteristics: IDDP is a highly scalable heuristic, column generation is a very flexible approach, and profile steering leads to very smooth demand profiles. We compare these algorithms with each other and with a lower bound solution in a large scale simulation case. For the demonstrated (deterministic) problem, both column generation and profile steering give near optimal results, within 1 – 2% of the lower bound. In a nested configuration, the approach has some scalability problems, yet the approach may scale well enough if the hierarchical structure is not too deep and the iteration count can be kept low. A local optimization that takes the needs of the higher level problems into account is a key technique to avoid unnecessary communication. We argue that the control step should not be completely separated from the planning step, but should rather be considered as the interface by which the planner can control the behaviour of devices. We embrace this approach in Chapter 4.

6.1.3. COORDINATION IN SPACE

Chapter 4

Flexible devices are a major source of uncertainty themselves, because their operation depends on human behaviour, which makes the forecasting of available flexibility for specific devices difficult. Dynamic dispatch approaches address this uncertainty by exploiting the interchangeability of devices, where we decide just-in-time which specific devices are going to be used, e.g. with a flexibility auction. Although this dynamic dispatching makes the approach more robust against disturbances of individual devices, it also makes the reasoning about the behaviour of the system more difficult for the planning. We propose a method to plan such a system based on the simulation of the dispatch process, where the planning result determines the configuration of a controller. More specifically, the planning determines the supply offer curves over time of an agent that controls the supply of electricity to a group of devices in a double sided single commodity Walrasian auction. As this leads to a planning problem that has a very complex internal structure, we use a metaheuristic optimization approach to address this problem.

We evaluate the method with a subset of Flex Street (10 heating systems), and find that the method finds results within 2 – 10% of the lower bound, depending on the considered configuration. This approach gives robust results even with large forecast errors and a small number of devices. However, the approach may still be improved in terms of efficiency and effectiveness by making better use of the available information from the simulation. Also, the control interface constrains the performance that can be achieved, in particular when limited local information is taken into account. For example, for the considered auction problem this means that the devices need a suitable bidding strategy to be able to fully exploit the flexibility that is available in the devices. This again requires that devices have knowledge about their own available flexibility.

6.1.4. COORDINATION IN PRACTICE

Chapter 5

There are some practical barriers for the uptake of DSM at a household level. One of these barriers is a lack of standardization of the interface to flexible devices, leading to high software development and maintenance costs. Control methods differ in their perspective on flexibility, which makes it difficult to find an useful common ground. Therefore, the *energy flexibility interface* (EFI) proposes to communicate the structure of energy flexibility (a description of the available control freedom) instead of a specific perspective on flexibility (e.g. price optimization or auction bidding). To describe the flexibility, EFI defines four control space classes for different types of devices: the *Uncontrollable*, *Time Shiftable*, *Buffer* and *Unconstrained* class. We develop a comprehensive TRIANA energy application prototype that implements the EFI on top of the *Energy Flexibility Platform and interface* (EF-Pi), a Java/OSGi environment for the development of energy management applications. To adapt TRIANA to this platform, we have revised the techniques in Chapter 3 to support multicommodity problems and problems with nonuniform time interval lengths. Furthermore, we introduce a *dynamic programming* (DP) formulation for the local optimization problems of each device flexibility control space class, using various techniques to make the evaluation of these DPs efficient (e.g. precomputation, approximations and Fourier-based convolution). The prototype supports the decentralized planning and control of real devices on low cost embedded hardware, and demonstrates that the techniques from Chapter 3 are feasible in an externally given framework (developed mostly independently by others). However, the resulting approach can not yet be seen as a complete solution, since it does not yet resolve the more conceptual challenges of Chapter 3. The prototype also shows that EFI supports multiple perspectives on energy flexibility in addition to just-in-time auction based methods.

6.2. CONCLUSIONS

The central problem addressed in this thesis, as formulated in Section 1.1, is:

How can we effectively and efficiently coordinate the flexibility in smart grids in time and space?

We have investigated two possible approaches to address this problem: a planning centric approach (that focuses on the time dimension), and a dispatch centric approach (that focuses on the space dimension). Both of these approaches give promising first results, yet still have significant issues both in efficiency and effectiveness. On the one hand, the time centric approach appears to be a better fit with existing work on the optimization of large scale systems. On the other hand, the space centric approach better allows to exploit the diversity between devices and appears to be more robust. In both cases, looking ahead and coordination between devices are the key ingredients for the optimized control of the available flexibility.

In more detail, we consider the answers for the subquestions separately:

- *How can we effectively and efficiently optimize the behaviour of large groups of devices in time and space?*

To efficiently optimize the behaviour, we first separate the time and space dimensions of the optimization problem, solve the problem in one dimension, observe how this affects the other dimension, and then modify the first problem to account for the observations in the other dimension.

In the time centric approach, decomposition is the main principle to keep the optimization tractable as the system is scaled up. A nested iterative optimization approach has scalability limits, but still scales well enough if the iteration count can be kept low enough. To keep the system scalable, the problems at the bottom of the hierarchy should already try to contribute to the objective of the problems higher in the tree within their local optimization. We make the optimization effective by the use of more sophisticated planning techniques that can give near-optimal results for large deterministic problems.

The space centric approach relies on a Walrasian auction approach (comparable e.g. to PowerMatcher) to optimize the just-in-time decisions of devices, which is a well-accepted highly scalable approach in the smart grid world. To address the time dimension, we determine a supply offer strategy that aims to optimize the behaviour of the devices over time, based on the simulation of the strategy. Even though a space centric approach makes the optimization robust against individual device disturbances, devices still have to consider their own local flexibility over time to dispatch devices in the right order.

- *How does uncertainty affect efficient and effective coordination, and how can we cope with this uncertainty?*

For the time centric perspective, the results presented by Figure B.3, in [HT:2], and the experience with the implementation from Chapter 5 suggest that a deterministic optimization approach leads to poor control results, depending on the nature of the forecast errors and on the way that we respond to them. Replanning gives a partial solution to these problems. Possible ways to address uncertainty more generally are stochastic optimization, robust optimization and reserve modelling, although these may come at the expense of extra computation and communication cost, and have not yet been investigated.

A space centric perspective on the problem with just-in-time control decisions is a promising strategy to reduce the impact of uncertainty, as the law of large numbers helps to make the problem less uncertain. Furthermore, late decisions make sure that the devices that we propose to use are actually available. A small number of scenarios appears to be sufficient to approximate the available flexibility at an aggregate level. Still, an aggregate perspective does not completely remove the need to consider specific devices, as many problems are inherently local (e.g. which heat buffer should be charged). Also, diversity is less effective against correlated problems (e.g. *photovoltaic* (PV) feed-in is highly correlated), so we should still ensure that the analysis covers all relevant cases. To make the approach more robust over time, we use a sloped supply offer curve. Although the efficiency may still be improved, we find that this approach effectively addresses uncertainty.

- *Can we use these optimization methods in a practical context?*

We develop a comprehensive prototype implementation of TRIANA in Chapter 5 based on an existing EFI, using the optimization methods from Chapter 3. EFI proposes to communicate the structure of energy flexibility instead of a specific perspective on flexibility. The abstraction of devices into control space classes by EFI enables portable device drivers, and thereby significantly reduces the effort needed to develop and maintain DSM implementations, as these drivers make up the largest part of these systems. This abstraction gives complex local optimization problems for each control space class, but now we only have to solve a single optimization problem to support a complete class of devices. To support EFI, we have implemented a DP optimization approach for each of the control space classes, using various techniques (e.g. precomputation) to make the evaluation of the DPs efficient. This leads to an energy management application that can decentrally optimize and control groups of real devices using third party device drivers on low cost embedded hardware. This demonstrates the feasibility of both EFI and TRIANA in a more practical environment, although the earlier subquestions have still left open some of the more fundamental challenges, such as how we should efficiently respond to intermittent changes in demand.

6.3. CONTRIBUTIONS

The main contributions of this thesis are:

- A composable and extensible *framework* for the hierarchical, temporally coupled optimization of DSM; Chapter 3

The framework builds on the ideas of [40], which in turn builds on ideas from conventional power system optimization;

- Efficient and effective optimization *algorithms* based on this framework;
 - * *Iterative distributed dynamic programming* (IDDP) gives a scalable yet less effective method (from [40]);
 - * Column generation is more effective and allows the more specific modelling of objectives (extends the concept of [40]);
 - * Profile steering gives very smooth demand profiles as it can exploit the flexibility at a fine granularity (introduced in [AT:6]);
- Support for *multicommodity* energy streams in the optimization;

The multicommodity dimension can be addressed in a way analog to the time dimension;
- A comparative *evaluation* of these algorithms against each other and a lower bound solution;

This tells that we have found a near optimal solution in the considered simulation case.

- An optimization *algorithm* for a Walrasian auction based coordination method based on the simulation of the dispatch process that is robust under uncertainty; Chapter 4

The algorithm (set) is based on metaheuristic optimization, and simulates the response to a dispatch strategy in several scenarios. The dynamic, just-in-time dispatching makes the approach more robust in space (between devices). We use a sloped supply offer curve to make the approach more robust over time.

- A proof-of-concept DSM system *implementation* using the algorithms of Chapter 3 on a practical management platform for energy flexibility; Ch. 5

- A mapping of TRIANA onto the EFI concept;

TRIANA adapts well to the EFI concept. EFI has motivated several extensions to the optimization framework, including multicommodity support and support for nonuniform time intervals;
- Efficient DP formulations for the control space classes of EFI;

We developed a tractable DP formulation for the complex *Buffer* control space class. For the *Time Shiftable* class, we found an elegant formulation that exploits Fourier-based convolution;
- Physical control of several devices in multiple set-ups;

The implementation has been applied for small scale experiments in three research projects (HEGRID, VIOS and e-balance).

6.4. RECOMMENDATIONS

Smart grids and DSM pose challenging problems, leaving ample opportunities for future work. Many aspects of these problems should still be resolved to make DSM the ideal “storage” resource that it promises to be. In the following, we outline some of the challenges that lie ahead.

6.4.1. BETTER OPTIMIZATION METHODS

Although the optimization methods have vastly improved over the course of this thesis, we have the impression that there is still room for improvement.

For the time centric perspective, we should consistently reduce the iteration count of the search methods to make the approach scale up to a national level. A further development of local optimization techniques that already account for the needs of the higher nodes appears to be a promising strategy. As discussed in Section 3.7.1, the lack of accounting for uncertainty makes the current approach less suitable in practice. Possible adaptations to this are uncertainty aware optimization techniques such as robust optimization, stochastic optimization, and/or reserve modelling. Note that we should take care to keep the extra effort resulting from the use of these techniques at an acceptable level. A part of a possible solution may be to make some of the flexibility at a lower level directly available to be dispatched at a higher level. Also, just-in-time dispatch methods may be useful to dynamically reoptimize the system, although it is not yet clear how this should be accounted for within the planning problem. Despite that we have focused on large optimization problems, we find that small (e.g. household level) problems come with their own challenges that should be addressed with a more specific approach.

For the space centric perspective, we may significantly reduce the computational effort of this method by extracting more information than just the objective value from the dispatch simulation. Initial experiments show that we may determine the (expected) bidding curves over time from the simulation, perform optimization based on these curves (similar to [265]), and then iteratively refine the estimate.

6.4.2. DIFFERENT TIME SCALES

The current approaches do not yet take into account the operation on different time scales very well. The time centric approach looks ahead over the coming period, but pays less attention to the present. On the other hand, the space centric approach looks just at the present, and pays attention to the future only implicitly. Both approaches have a limited reasoning beyond their optimization horizon, e.g. over seasons. At the moment this may not be a problem since most current storage techniques address only the short term. However, long term energy storage over the seasons is likely to become an even more important challenge. This leads to a balancing problem over a much longer time horizon than that we address in this thesis. Therefore, a suitable integration between different time scales remains an important open issue.

A specific example where this integration is relevant at short time scales is the feed-in pattern of PV on cloudy days. On such days, the supply profile over time can exhibit a high degree of jitter, alternating between full feed-in and no feed-in at

a minute scale, and we may want to dynamically switch between operating modes to compensate for this jitter. Note that these problems still occur at a neighbourhood level due to the highly correlated nature of PV in the same location. It may be possible to represent these interactions at a sub time interval scale in a planning problem with a *continuous time Markov chain* (CTMC) model, similar to how we addressed state transition lockout timers in Section 5.4.4.1.

6.4.3. SECURITY AND DEPENDABILITY

The discussion on robustness has so far focused primarily on the energy domain, i.e. whether we can cope with uncertainties in demand. However, in practice a reliable energy supply relies on more than just having sufficient energy available.

The security of the smart grid is an obvious concern that has already attracted the attention of security researchers. Residential DSM gives an enormous attack surface in particular, as it opens millions of access points to the smart grid that can in practice not be physically protected substantially. Containment and detection are important techniques to limit the damage from such events.

The dependability of the system is important as well, as the smart grid may become an integral and thereby critical part of the operation our complete infrastructure. Otherwise, we still need heavily overdimensioned hardware (cables and transformers) to support the worst case without control, or even the more difficult case with adversary control (by an attacker). Redundancy and fail-safe techniques can help to keep the infrastructure available, possibly with reduced functionality. Still, even if the components appear to work as intended, “blind trust” at the application level may seriously impact robustness: single bit errors – which do not occur just on communication channels – or programming errors (e.g. due to numerical problems) in any connected device may present extremely low or high demand values, which may lead to a cascading failure. Simple sanity checks can already prevent most serious problems: for example, a household will never consume 2147483647 W. Similar to the earlier example, we have seen that integer implementations of optimization algorithms may inadvertently overflow if we do not control the domain of the demand and price values, which means that an extremely expensive solution is suddenly considered as an extremely cheap solution. These problems are hard to detect at an aggregate level, because the optimization from the other devices diligently masks these errors. More advanced monitoring and statistics may uncover and sometimes recover more subtle problems (e.g. degraded measurement hardware, software errors or cheating).

Most DSM control algorithms rely on honest, cooperative agents. While we presume that most agents will cooperate, cheating attempts are inevitable in a large scale deployment if there are incentives to do so. These incentives may be directly monetary (e.g. do not pay for what is used), indirectly monetary (e.g. reducing wear on your own devices at the expense of others), or immaterial (e.g. charging an *electric vehicle* (EV) earlier than others). A game theoretic approach may help to keep users honest, and help to ensure that the interests of the customer remain aligned to the common interest.

The scale of the DSM problem gives a lot of places where things can go wrong, but also gives a lot of opportunities to succeed in spite of this. There is *massive redundancy* in the households and the devices: at a neighbourhood level, it should be virtually irrelevant if someone's battery breaks (however, devices should still not fail too often to avoid excessive maintenance costs). This redundancy also provides opportunities for distributed control: the group planning problems are computationally inexpensive enough to be replicated several times across a neighbourhood. For communication, overlay networks, e.g. based on distributed hash tables, may give a scalable and robust communication facility between the households. Note that this organization may also reduce the communication burden on a (de)centralized controller.

6.4.4. GUARANTEED PERFORMANCE AND CAPACITY PLANNING

Decentralized smart grid control shows the potential to dramatically decrease load on the electricity infrastructure, as well as on energy resources in general, which is a well-known result. By this, the life span of these resources may increase. However, for capacity planning, a grid operator needs a *guaranteed* load reduction (with a certain confidence level) that can be predicted in advance, before the infrastructure gets adapted. Current approaches rely on simulations to estimate the worst case behaviour of a cluster of devices, which take substantial effort in computation and in modelling, and are unlikely to find the real worst case scenario. Consequently, a (prudent) grid operator can not rely on this result in capacity planning; the operator may even need more resources in comparison to business as usual, as the usual statistical diversity design rules no longer hold.

If we have control over the available DSM devices, we may use a similar reasoning to determine the number of smart devices that have to be installed in a neighbourhood to reach the desired demand response capability. In considering the devices to use, an aggregate flexibility model as used in Appendix B.6 may be helpful to make sensible investment decisions, such that the system can offer power and energy flexibility where it matters the most over the day and over the year (this may be posed as an optimization problem similar to power generation expansion planning, see e.g. [154]).

6.4.5. FORECAST WHAT WE CAN

Currently, the optimization approach assumes that we are able to make a perfect forecast of all the parameters of the problem. However, this is a quite unrealistic assumption on what information is available and on the quality of forecast methods. One argument for a poor forecast may be the uncertainty that is present in the considered situation (e.g. the weather). Nevertheless, it is important to look at all useful information that can actually be obtained at low cost from installed (or, in the future, realizable) devices, and find ways to interpret this as useful information within a planning. In this context, it is also important to look at the value of specific information for the planning: it may be far more important to have an accurate view on the amount of energy that we have to deliver in a certain period than to know the exact demand profile. Furthermore, information on how inaccurate the forecast is (which changes over time) may be helpful.

6.4.6. COST REDUCTION

The presented practical DSM approach still asks quite a lot from the environment, making it expensive in terms of cost and user effort. The cost related to the control system itself (installation, operation and maintenance) may eat up the profit that we gain from the energy optimization. For example, a smart washing machine should cost less than €50 more than a regular one (and preferably less),¹ rather than the €500 that it costs more today. Note that more potent sources of DSM flexibility, such as EVs, may dilute the value of flexibility, i.e. the cheapest available sources that cover the flexibility needs determine its value (according to a merit order principle), as flexibility has little intrinsic value if we never use it.

The same reasoning on cost reduction holds for the control system at a household level itself. As device controllers are growing increasingly powerful, it may be worthwhile to consider leaving out the household controller altogether, and implement its functionality as a distributed system within the devices, similar to what we sketched in Section 6.4.3 (where the household controllers together act as a surrogate for a neighbourhood controller). We also see a trend that smart devices communicate directly to a cloud server for smart home applications while skipping the household level, e.g. to be able to turn up a thermostat from anywhere in the world on a mobile phone. It may be possible to make a low cost implementation of DSM on top of this functionality, although it should be clear that this increases the risk to lose control over these devices.

If we truly want to make DSM the low cost and efficient “storage” instrument that it promises to be, then it should be implemented at a low cost, yet also realize an efficient and effective coordination of the available flexibility.

1. Assuming e.g. that we can save €0.05 per day by shifting the demand, e.g. from PV self-consumption, we save €18.25 per year, and thereby have a reasonable payback period of 3 years.

THE WORLD OF ENERGY

energy transition 239 · smart energy 250

• • • • •

THIS appendix serves as a basic introduction to the energy transition and smart grid domains, e.g. for engineering students with no prior background in these themes. We continue with the corresponding coordination methods that we need for control in Chapter 2.

A.1. ENERGY TRANSITION

In this section, we discuss the trends that are currently changing the energy landscape: renewable generation (Appendix A.1.1), the electrification of demand (Appendix A.1.2), decentralization of generation (Appendix A.1.3), and energy storage (Appendix A.1.3). These trends are summarized by the term *energy transition*. The ambition of the energy transition is to make the entire life cycle of energy – production, transportation, storage, and consumption – clean and renewable, such that the use of these resources will not harm or restrict future generations in the long run (i.e. centuries).

The previous pivotal point in energy policy, the 1973 oil crisis [151], shaped the landscape of the current energy transition. During this crisis, it became painfully clear that (access to) energy is a limited, vital resource that should be economized. Next to immediate temporary energy conservation actions (e.g. lower speed limits and car free Sundays), the crisis triggered a switch from oil to nuclear, coal and gas sources, the set up of strategic long-term energy reserves, and started initiatives to improve energy efficiency and to exploit renewable energy sources. Many of the major energy institutions were founded (e.g. the IEA, DOE, and NREL in the US) or retargeted (e.g. ECN) in the mid-1970s.

A new driving factor for the current pressure on changing energy systems to renewable sources is environmental awareness. One of the environmental hazards of using fossil fuels as an energy source is the emission of the greenhouse gas *carbon dioxide* (CO_2) into the atmosphere, which is generally believed to lead to global warming. Global warming increases sea levels and leads to extreme weather conditions [227]. Most industrialized nations have committed by treaty (e.g. the 1997 Kyoto protocol [329]) to large reductions in greenhouse gas emissions. The *European Union* (EU) has the 20–20–20 target, which aims for a 20% reduction in

FIGURE A.1: Development of residential consumer retail electricity prices and solar PV feed-in tariffs over the last decade [48, 330, 360].

(CO₂ equivalent) greenhouse gas emissions and a 20% share of renewable electricity generation by the year 2020 compared to 1990, or targets of 30% if other developed countries commit to comparable emission targets as well [111, 321]. Similar, higher targets are set for 2030 [108] and 2050 [109]. So far, the progress towards these (unevenly distributed) targets varies between member states [110]; e.g. the Netherlands is lagging behind significantly as of 2012 (see also [89: p. 80]), which has lead to the Energieakkoord response plan [301].

Renewables have vastly different characteristics than conventional energy sources, which causes system challenges. In the following, we elaborate on the challenges that arise from the energy transition. These challenges justify the smart grid approach, which we describe in Appendix A.2. Readers without a cursory knowledge on electric power systems are encouraged to review Appendix A first.

A.1.1. RENEWABLE GENERATION

Renewable generation, using *renewable energy sources* (RES), is a key element in the energy transition (these two terms are used interchangeably). Current renewable generators behave vastly different from conventional generators. Consequently, a “fit and forget” integration of renewable energy sources in the electricity grid is only workable as long as the share of renewable generation is small enough to be considered as a disturbance to conventional generation.

In several countries in the EU, substantial amounts of renewables have been integrated by stretching the capabilities of (mostly) conventional generation and transportation techniques. For example, as of 2014, Germany has a renewable share of 31% for the electricity sector, using a mix of PV, biomass and wind generation [49], and Denmark even has a wind share of 39.1% as of 2014 [192].

The integration of large amounts of renewables seems to have a large cost for the consumer. Figure A.1 shows that the residential electricity retail prices in countries with a large share of renewable generation have increased substantially over the last years, whereas prices have remained almost constant in countries where the

renewable share is relatively low (e.g. the Netherlands). This is partially due to the energy policies that have made renewables popular in these countries: generous incentives and attractive feed-in rules and tariffs.

Next to the direct costs from these schemes, there are also system costs, and costs for external parties. At the moment, the high penetration of renewables in a small number of countries is possible by participation in the European electricity system and in European electricity markets. This allows these countries to sell the peak production of the renewable generation, and offset shortages of renewable generation by buying electricity on the integrated market. Unfortunately, the production profiles of specific types of RES tend to coincide in neighbouring countries, which nowadays already leads to *negative* market prices during the afternoon in summer due to excess PV feed-in [199]. When in the future the renewable production increases in most countries, this effect is expected to occur more frequently, making it harder to achieve renewable shares as in Germany and Denmark in the same way.

Competition from RES, or even prioritizing RES, also means that conventional producers can sell less electricity, and that the market clearing price will decrease. Furthermore, under the influence of RES, the requested production profile changes: rather than a constant base load pattern, conventional generators should deliver the intermittent pattern that is not met by the renewables. Both developments lead to power plants that make far less full-load hours than necessary to be profitable. Nowadays, even nuclear power plants – the pinnacle of base load generation together with coal power plants – can operate in a load following mode (although this capability is used sparingly for economic reasons) [213]. Consequently, the large conventional power plant operators will make substantial losses, as is already the case in Germany and across Europe nowadays [106, 280]. Some gas plants in Germany are currently used for only hundreds of hours per year [102], and many plants are mothballed, decommissioned [10], or even disassembled and shipped to more lucrative markets with less competition from renewables [261]. In the Netherlands, in particular gas plant operators have similar problems for different reasons, i.e. the introduction of large new coal plants while the old plants have not yet closed, high gas prices, and dumping of foreign excess renewable energy [1, 89, 261, 287, 301, 325]. Note that natural gas prices in the US have plummeted due to the extensive exploitation of shale gas, leading to renewed interest in gas fired generation [168].

Whereas the closing of inefficient, unprofitable conventional plants would be appropriate, some of these plants are still necessary for system stability. To ensure security of supply, new capacity mechanisms are introduced to keep these plants operational [328]. While this approach ensures security of supply, it is very expensive.

As a further example of the system efforts that are needed to increase the fraction of renewable generation to a substantial level, we consider Denmark, which has integrated large amounts of wind generation in their power system [202]. For balancing, the flexibility of existing *combined heat and power* (CHP) district heating plants is used [9, 302]. Also, modifications to existing coal power plants allow for fast ramping and a very low (10 – 20%) minimum output [64, 223]. Furthermore, heavy investments in interconnection with Europe, and in *high voltage DC* (HVDC) lines to Norway – which has abundant hydro reservoirs – give access to both the

Scandinavian and European electricity markets. Reverse power flows towards Norway reduce the use of hydroelectric power for base load generation, which conserves reservoir water for flexible generation [113].

These examples show that it is very important to consider the different properties and potentials of the different sources of renewable generation, as well as the geographical and political context. In the following, we address the most common renewable generation sources.

A.1.1.1. PHOTOVOLTAICS

Photovoltaics or PV uses sunlight to convert solar radiation to an electric *direct current* (DC) flow, which can be injected on the electricity grid with an inverter. The power production profile depends on the used technology, the orientation of the panels (which may be dynamic by mechanical rotation of the array), and on weather conditions.

PV has earned a high popularity in the residential sector, due to government subsidies, high residential energy taxes (which also act as a kind of subsidy), and plummeting manufacturing costs. After a period of explosive, exponential growth, it is increasingly acknowledged that the subsidies have been “too successful” [215]. The large scale introduction of PV in the *low voltage* (LV) distribution grid has reached the point that the load curve over the day has distinctly changed in shape, which nowadays leads to a “camel” curve (i.e. with a gap in the usual day–night load profile around noon), and is expected to lead to a *duck curve* in the future [54] (we will show the load profile of a household that exhibits a duck curve in Figure A.2). Also, weather conditions are highly correlated and localized, in particular for small areas with similar, synchronized cloud coverage conditions; at short time scales, this locally leads to a jittered feed-in profile on partly cloudy days; consider e.g. [67: Figure 2]. Whereas most PV plants are connected on the LV grid at households, utility scale installations are common as well, which are connected at an *medium voltage* (MV) level [67]. All solar generation systems have a seasonal variability, where production is high in summer and significantly lower in winter. For an example on the seasonal impact of PV on the German electricity balance, see [255: pp. 20, 22]. We refer to [93] for an overview of the technical system effects of PV, and [364] for an overview of the economical effects of PV in Germany.

The Dutch energy law uses net metering (*salderen*) for feed in up to 5000 kWh/year, and a supplier specific “reasonable compensation” for production in excess of this threshold [240]. This means that a consumer can “store” energy on the grid in summer, and then use this energy credit to pay for demand in winter. In the *Stroomversnelling* agreement, 111000 series-built council houses from the 1950s–70s are renovated to have a zero annual net energy use, using energy saving technologies (e.g. modern insulation) and “sustainable sources from within 10 km of the house [on average]” [311: p. 7]; in practice, this is often implemented using net metered PV at a household level and air source heat pumps [312], which leads to local load concentrations and PV feed-in peaks [96, 308]. It is anticipated that net metering in the Netherlands will be limited or removed in 2017 – 2020, leaving only the reasonable compensation for excess production [298].

PV is not the only form of solar energy (in fact, the sun fuels almost all renewable energy processes). Notable examples of other solar renewable generation techniques are solar thermal collectors (which heat up water directly and store it in a buffer) and *concentrated solar power* (CSP) [357]. CSP uses a large array of mirrors to heat up a central collector, which can run a thermal generation process. With a high-temperature molten salt heat buffer, or an auxiliary fuel backup system, the production can be sustained after sunset; longer term (interseasonal) storage is under development [376]. The units are often very large in power (e.g. 50 – 500 MW), have a high land use, and need vast amounts of cooling water.

A.1.1.2. WIND

Wind power uses generators with wings mounted on a tower to produce electricity. The production depends on local wind conditions, and thereby on the siting and height (higher is better); furthermore, larger units in general have a higher capacity factor. Onshore and offshore wind have vastly different economics: onshore wind has a capacity factor of 10 – 30% [36, 173, 247: p. 44], whereas modern offshore wind parks have a capacity factor of 30 – 50%; however, offshore wind gives transmission and maintenance challenges due to their remote, harsh (and windy) locations [180]. The typical unit size is 2 – 5 MWp, which are aggregated in wind farms of e.g. 150 MWp. Both onshore and offshore wind are subject to *not in my backyard* (NIMBY) resistance, which pushes wind parks away from load centers.

Wind energy is seen as a complementary source to PV, because the production over the year follows the opposite average pattern: whereas PV is strongest in summer, wind production is higher in winter. With an appropriate scaled dimensioning, the production of these sources can be balanced on a seasonal level [364: p. 40]. Whereas PV is characterized by a daily feed-in peak, wind gives longer periods (e.g. a week) of high or low intermittent production; consider e.g. [364: Figure 58].

A.1.1.3. BIOGAS

Biogas produces hydrocarbon gases (mostly methane) from the digestion of organic matter by bacteria, for example from landfill gas, waste water treatment, or agricultural waste products (in practice however, the agricultural “waste” is produced expressly for this purpose [43]). The gas can be transported, or “upgraded” and injected into the natural gas grid; however, the gas is typically converted to electricity with an on-site internal combustion engine and generator. The average unit size is 500 kW for an agricultural installation, and are usually connected at the MV distribution grid. Biogas installations have a very high utilization factor (over 90% [251]).

Whereas biogas plants have the potential to provide flexible generation, investment costs and process stability motivate a flat, full-load production profile; to rectify this, incentives have been introduced in Germany that reward dispatchable, flexible generation [144].

A.1.1.4. HYDROELECTRIC POWER

Hydroelectric power plants provide cheap, clean, and dispatchable power, and have thereby formed a valued component of electric power systems since the beginning of the industry. Hydroelectric plants rely on water streams, and thereby need specific geographical resources: height differences and a streaming water supply. In countries that do have these geographical resources, hydroelectric power is an integral part of electric power production, and system balancing in particular.

The most dominant type of hydroelectricity uses a dam to make an artificial lake, which can store large amounts of water. An artificial lake gives access to a consistent, controllable water flow that allows for energy production throughout the year. However, a lake relies on a natural feed-in of water (from rain and melting ice), which has a seasonal pattern and is subject to “wet” and “dry” years. Consequently, it may run empty, which affects power markets significantly: in 2010, depleted reservoirs contributed to a series of factor 15 – 20 price spikes on the Nordic market [82]. Furthermore, multiple lakes often use the same water stream in cascade, leading to multi-reservoir optimization problems (i.e. the high lake needs to produce power first before the connected lower lake can be used) [201].

A dammed lake may also be used to store excess electricity production, by pumping water back up from the lower end of the dam, which is called *pumped hydro*. Pumped hydro is a major storage resource in current operation with e.g. a 60 – 80% efficiency [45], and allows for a more intensive use of the generation capacity. Hydro and pumped hydro power resources are often considered explicitly in power system optimization problems, as these behave vastly different from thermal generation units (a hydroelectric plant does not have a thermal commitment, but does need long term storage management); consider e.g. [201, 228, 234, 310] (see also Section 2.3.1.3.1).

A.1.2. ELECTRIFICATION OF DEMAND

Electrification is the second key element of the energy transition. More and more devices switch to an energy supply based on electricity. Two main examples are electric vehicles and heat pumps, which we will discuss later in this section. Furthermore, most RES supply energy through the electricity grid. Electrification enables a flexible allocation of the primary generation resources to the loads, as electricity provides the ability to transparently switch between conventional energy sources and renewable energy sources. This ability provides a gradual path towards a full transition to renewable energy sources. However, it also brings challenges, as the transport of energy shifts from moving fuels to moving electricity. Most fuels have a very high energy density: for example, 1 ℓ of diesel fuel contains 10 kWh of thermal energy [359]. Furthermore, solid and liquid fuels can be easily stored, in contrast to electricity. In most places, the grid is not designed to cover the complete energy demand; electricity represents only a fraction of the total energy demand [89]. Consequently, electrification of the supply and demand brings challenges in energy logistics, and will have a major impact on the operation of the LV distribution grids in particular [337].

FIGURE A.2: Example of the effect of electrification on the electricity demand profile in a Dutch energy neutral house (nulwoning), March 8th 2015 [288: personal communication].

To appreciate the impact of electrification, consider the electricity load profile of an all-electric existing house with a heat pump and an EV in Figure A.2. In this figure, most of the morning demand results from the use of a heat pump, and the evening peak results from EV charging. An average (non-electrified) Dutch household consumes about 10 kWh of electricity per day in 2014 [89]. In this example (in spring, 2015), the morning run of the heat pump consumes about 3 kWh, and the EV charging consumes 20 kWh, next to 22 kWh of PV production and 10 kWh of miscellaneous load. Although the net consumption (11 kWh) is slightly more than in a normal household on this particular sunny day, it is evident that electrification gives significant changes to the shape of the load profile, and the transport of electricity on the grid. In winter, heat pump consumption is higher and PV production is lower, leading to a significant increase in net consumption as well. The increase in load on the grid requires extensive, costly grid reinforcements in a business as usual approach, as the distribution grid is dimensioned for a lower estimated concurrent peak load.

On the other hand, electrification brings new opportunities to decrease energy use. The already mentioned examples are EVs and heat pumps, which are usually much more efficient than their conventional, fossil-fired counterparts. In these applications, the constraints of electricity transport and storage motivate efficiency improvements: for example, whereas an inefficient fuel-based car is expensive to operate, an inefficient EV needs a large battery, and thereby becomes heavy (which gives poor vehicle handling), takes long to charge, and strains the grid. Heat pumps have received considerable efficiency optimization effort to reduce their load on the grid [372]. Furthermore, many energy saving innovations are found in the electric domain: for example, the introduction of high efficiency fluorescent lighting (TL in Dutch) and *light emitting diode* (LED) lighting contributes to a steady decline in energy use (e.g. 24% over the last decade in the UK [327]), in spite of an increasing demand for lighting [326].

In the following, we discuss two examples of electrification that have a large impact on distribution grids: heat pumps and electric vehicles.

A.1.2.1. HEAT PUMPS

Heat pumps are heating devices that do not produce heat directly, but instead force heat to move from one point to another. A thermodynamic process (*refrigeration cycle*) extracts heat from a cold source and releases it at a hot sink – the opposite to natural heat flow. The process exploits the condensation and evaporation of a working fluid, which circulates between the two ends. Refrigerators and *heating, ventilation and air conditioning* (HVAC) units also use the heat pump principle. In some configurations, the operation of the heat pump is reversible, which allows both heating and cooling. For a further discussion on heat pump principles, see e.g. [319, 344].

Heat pumps come in many configurations, with vastly different characteristics. Heat pumps are often more energy efficient than gas heating, whereby the difference depends on the concrete power plant efficiency (typically 30 – 60%). The *coefficient of performance* (COP) describes the efficiency of a heat pump, which gives the ratio between the effective heat transfer and the electricity use (higher is better). The COP strongly depends on the system configuration, and on the temperature difference between the source and the sink, i.e. a low temperature difference gives a high COP. The most common types of heat pumps are air source and ground source heat pumps. Many heat pump systems have a high-power auxiliary resistance heating rod to meet peak demand; these heating rods have a “COP” of 1 for heating.

Air source heat pumps have a relatively low investment cost and are easy to install, but commonly reach a COP of only 2 – 3. Due to the dependence of the COP on the outdoor temperature, cooling is inefficient on hot days, and heating is inefficient on cold days – precisely the opposite of what is desired. On very cold days (below $\approx 4^\circ\text{C}$), the resistance heater is more efficient than the heat pump. Therefore, air source heat pumps are inappropriate for heating in climates where these low temperatures are common during the heating season. Retrofit HVACs often use air source heat pumps.

In contrast, ground source heat pumps use an underground loop of pipes, which provides a source or sink with a year round near constant temperature. By this, ground source heat pumps often reach a COP of 4 – 6 when combined with a low working temperature heat sink to make the temperature difference small, usually in combination with floor heating. However, this comes at the cost of building the ground loop, which is expensive and impractical for existing properties, and relies on a suitable local geology.

Modern energy efficient buildings are designed with good insulation and a high solar gain during winter (i.e. large, south facing windows). Hereby, these buildings have a very low heat demand, which allows a single house to use a small ($5 - 10 \text{ kW}_{\text{th}} \rightarrow 1 - 5 \text{ kW}_{\text{el}}$) heat pump. Thermal buffering (or exploiting the slow heat loss of a building, combined with hot water tank storage) avoids the need for spikes in heat generation, and thereby avoid the use of auxiliary (usually resistance) heating.¹

1. Some heat pump systems always use resistance heating to produce hot tap water, or use it to heat up water from the heat pump to the desired high temperature.

While heat pump based electrified heating may be more efficient than conventional, fuel-fired heating, the burden to deliver the energy rests on the electricity grid. The extra load is significant – especially on extreme cold days that force resistance heating if the heat pump is underdimensioned – and coincides with the morning and evening peaks in the rest of the electric demand. To avoid the extra peak from resistance heating, some utility incentive programs explicitly forbid resistance heating [15].

In hot climates where cooling is dominant (e.g. the Mediterranean, and the southern US), air conditioning heat pumps have a large share in residential and commercial electricity demand, and cause demand peaks in the afternoon in summer. In the southern US, HVACs with 3 up to 20 kW_{el} are common for single-family homes (e.g. [138, 356]) – and these run most of the time during hot summer afternoons. Therefore, increasing the efficiency of these units (i.e. by giving incentives to replace old units, and setting standards for new units) is a key priority in the US to reduce peak load [372].

Clusters of heat pumps have a significant impact on the load and the required design capacity of the electricity grid [3, 266, 267]. Several projects in the Netherlands have developed “all-electric” neighbourhoods where the usual gas grid is left out. The heat demand of households is highly correlated; consequently, the load of heat pumps without a grid aware control synchronizes, which leads to a low diversity factor. In one example, heat pumps tripled the annual load, yet increased the peak load dimensioning by a factor 5, increasing the investment cost of the electricity grid by 180% compared to business as usual [342].

A.1.2.2. ELECTRIC VEHICLES

Electric vehicles promise to bring similar dramatic efficiency improvements to transportation as heat pumps to heating and cooling. Car combustion engines have an efficiency of roughly 20 – 30%, and rely on fuels that need extensive preprocessing in oil refineries. In contrast, EVs can use stationary, shared, highly efficient power plants, as well as any renewable generation that is available at the time of charging, and use highly efficient (> 90%) electric motors in the vehicle. These motors are significantly lighter and smaller than combustion engines with similar power. Moreover, electric energy storage and the electric propulsion enable regenerative braking, which cuts energy use especially in city driving.

The first challenges with EVs are the driving range and up-front cost. Both of these challenges result from limitations in battery technology. Current batteries have an energy density that is 1 – 2 orders of magnitude lower than conventional liquid fuels [359]. In an EV application, this low density introduces a weight problem. To illustrate this point clearly, a fuel tank with 30 ℓ (25 kg) of diesel contains 300 kWh_{th}, which roughly yields 100 kWh_{eff}. A lithium-ion battery with the same energy content weighs at least 750 kg – by which it contributes roughly half of the mass of the vehicle [103]. At an automotive scale and reliability grade, batteries are expensive, costing \$400 – 800 /kWh in 2012, depending on the specific technology used [90]. Hereby, the 100 kWh example battery would cost \$40 000 – 80 000, which is a prohibitive up-front cost for the general public. However, a full recharge

at home would cost €22 (at €0.22/kWh) rather than €45 (at €1.50/ℓ for diesel fuel), based on approximate Dutch consumer prices. To cut costs and weight, the affordable EVs that are currently on the market have a small battery (around 20 kWh), and thereby have a limited driving range (≈ 100 km). This limited range results in range anxiety with consumers, similar to driving a conventional car with a near-empty fuel tank. As a compromise, *plug-in hybrid EVs* (PHEVs) combine a small (5 – 10 kWh, 20 – 40 km electric range) battery with a conventional engine, e.g. the Toyota Prius [323]. At the other end of the spectrum are EVs which do have a large battery in spite of the high cost and weight, most notably the Tesla Model S (42 – 85 kWh $\rightarrow$ 250 – 450 km) [103], which is highly popular among early adopters.² As battery technology advances, energy density and cost are bound to improve. By the year 2030, it is expected that the majority of new cars will be EVs, or at least PHEVs, in Europe and the US [24, 91].

As EVs with large batteries become more popular, at-home EV charging becomes an increasingly challenging problem for the distribution systems. As already illustrated in Figure A.2, even a fairly small amount of energy from an automotive perspective (i.e. 20 kWh) gives a tremendous increase in load from a residential electricity perspective. Furthermore, current chargers implement a greedy (“as soon as possible”) charging policy, which results in large, synchronized peaks in demand as EV drivers return home in the early evening. By this, even a modest concentration of EVs – which may already occur in rich neighbourhoods – can give major grid congestion [161], or even cause grid failures [208]. Therefore, EV charging is a specialized subtopic in smart grid research, and most utilities operate or participate in research programs that specifically address EVs, e.g. [101, 140, 184, 281, 314, 373].

A.1.3. DECENTRALIZATION OF GENERATION

The third key element of the energy transition is the decentralization of energy production, or *distributed generation* (DG). Large conventional power plants are concentrated at locations with e.g. access to cooling water (a large river or the sea), access to the *high voltage* (HV) transmission grid, and sufficient air quality [269]. In contrast, many renewable generators are a lot smaller and have different siting requirements than these large units (e.g. a non-overshadowed rooftop), and are therefore connected to the MV and LV distribution grid. Increasingly, residential consumers connect DGs at the household level, such as PV (Appendix A.1.1.1), microCHPs (Appendix A.2.1.2), and backup generators (Appendix A.2.2.1.3).

The introduction of large amounts of DG in the distribution grid gives various problems [247]. The conventional design of the electricity grid follows a cascade principle, where energy flows downward from the generators that are connected to the HV transmission grid to the loads on the MV and LV distribution grids; reverse flows are considered as a fault. This assumption was embedded in protection mechanisms, and in voltage management: if power only flows down, then the voltage within the grid can only drop; therefore, grid operators configure distribution transformers at the top of the operating range, which allows the use of the full

2. In Norway (2013 – 2015), the Model S is one of the best selling cars, which is attributed to tax exemptions and road privileges for EVs [361, 363]. The total EV market share in 2014 was 12.5%.

range (see Appendix A.2.2.2.1). However, DG with a net feed-in can violate this assumption, as power can flow upstream in the grid. Localized feed-in “hot spots” may occur that are invisible at the transformer level. Except for the backup generators, these DGs usually do not produce electricity to follow the load, but rather produce electricity whenever the source is available. Therefore, the production coincides with demand at some times, but may at other times produce power when there is no local demand, which results in net feed-in. Furthermore, DG units of the same type in the same environment tend to correlate strongly in production behaviour, which leads to synchronized feed-in peaks. Hereby, the variability of the (perceived) load profile increases.

Whereas the general trend in the energy transition from a global perspective is that more aspects related to energy – and in particular electricity – move downward in hierarchy, some resources at the bottom end of the hierarchy are moving upward. Cooperation and sharing help to increase the utilization of resources, which allows various services to be delivered at lower cost. For example, a high efficiency heating system, such as a microCHP or heat pump, takes a large investment that is hard to economically justify at an individual level; these technologies benefit from economies of scale (in the sense of unit size rather than volume), which motivates district heating (see Appendix A.2.1.1). Similarly, a shared backup power facility can be significantly cheaper than an individual backup source; in recent work, this is known as a microgrid (Appendix A.2.2.1.3). In residential and rural areas, local energy cooperatives have started, many of which aim for energy independence, using local RES [164] (however, a true autonomy from the central grid is still quite expensive and often relies on external fuels, and is thereby uncommon in places where a central infrastructure is available).

A.1.4. STORAGE

The final key element of the energy transition is storage. Renewables produce energy whenever the renewable supply of energy is available, which leads to variations in production. Furthermore, the production does not follow the load. To bridge the gap between supply and demand, the system needs flexible generation and energy storage. Storage can compensate for time varying differences in supply and demand by shifting energy in time. By this, the variable production of RES can match the demand. With distributed storage, locality can be exploited, which saves transport capacity, losses, and wear on the grid.

Current storage technologies have limitations and substantial drawbacks. By definition, a storage has a capacity, i.e. it becomes full and empty during use. To save costs, storage resources tend to be small in comparison to the expected load, or even underdimensioned intentionally. A storage resource can have constraints on the use of the resource, e.g. charge and discharge rates, ramp rates, time of use, and economical constraints on the way that the resource is used (e.g. trying to avoid switches between charging and discharging when the device is near empty). Furthermore, a storage resource gives an efficiency penalty: e.g. 10 – 70% of energy is lost by charging (depending on the technology), and there can be losses over time. Conse-

quently, flexible generation is nowadays still the preferred method to match gaps in renewable production; however, this approach is not sustainable with a larger contribution of renewables. We will further address storage in Appendix A.2.2.2.2.

As an alternative to these “explicit” forms of storage, literature proposes to exploit the properties of the demand itself to implement storage; we follow this concept, and we will elaborate further on this concept in Appendix A.2.2.3.2.

A.2. SMART ENERGY SYSTEMS

The energy transition (Appendix A.1) increasingly shifts energy supply and demand from the direct use of fossil fuels to the electric domain, with generation from renewable sources. Innovative (applications of) energy technologies can help to avoid restrictions and escalating costs in this transition, in contrast to a business as usual, capacity expansion driven approach. These applications are described as *smart energy systems* (SES) (in the Netherlands, the term *system integration* is also used, e.g. [79]).

While the SES description has emerged only in recent years,³ the concept already exists for a longer time. Many post-oil crisis energy efficiency techniques, such as cogeneration, district heating, and engine control units, would nowadays be labelled as SES. What is new, however, is the increase in affordability of *information and communication technology* (ICT)-based control methods, which makes control attractive for increasingly small energy resources, and the context of the energy transition which renews interest in these techniques. For example, a smart electric vehicle may help to relieve congestion on the electricity grid or reduce cost on the electricity markets by charging at appropriate times, or a hybrid vehicle may use the best (i.e. cheapest) available energy source. Furthermore, the domains that comprise SES are increasingly converging: a holistic system perspective on energy, where techniques from different domains complement each other, helps to find efficient solutions with existing technology, and helps to direct research efforts into new energy technologies. For example, a power to gas installation can convert surplus renewable electricity energy, which would otherwise be discarded, into a useful form that can be stored.

The following two subsections discuss two major orthogonal, yet increasingly converging lines of smart energy systems research. First, in Appendix A.2.1 we discuss hybrid energy systems, which exploit the differences between energy carriers. Second, in Appendix A.2.2, we discuss smart grids, which aim to improve the utilization of the infrastructure. While smart grids originally only considered the electricity infrastructure, the smart grid field increasingly also considers other infrastructures – in particular district heating grids. However, in this thesis we still focus mainly on smart grid applications that change the use of the electricity infrastructure in time. Many of these applications rely on the concerted actions of distributed resources, i.e. a group of devices that acts as a single large energy resource by aggregation. As coordination is an essential aspect to enable such an aggregation, we continue with this aspect in Section 2.3.

3. As a generalization of the popular term *smart grids*, which we discuss in Appendix A.2.2.

A.2.1. HYBRID ENERGY SYSTEMS

Different types of energy carriers have different properties. For example, in comparison with electricity, heat is easy to store in large quantities, but hard to transport over long distances. As a consequence, a smart energy system might therefore transport electricity, convert this electricity into heat, and store this heat. Others may also argue that heat storage has losses, and opt for electric battery storage instead, which has different loss behaviour. For hybrid energy systems, it is a challenge to choose appropriate combinations of energy technologies and energy carriers to avoid losses, or if this is not possible to use the losses to perform useful work in a different process (e.g. use waste heat for residential heating). Whereas conventional hybrid energy systems generally choose a static hybrid configuration to save investment costs, SES that allow the ability to switch between energy sources according to current conditions have a higher potential and are an interesting direction for research, which has already been implemented in some systems.

We present two practical, widespread applications of hybrid energy systems: district heating and cogeneration. Another widespread application of hybrid energy systems are PHEVs, which have been shortly discussed in Appendix A.1.2.2.

A.2.1.1. DISTRICT HEATING

Many industrial processes use vast amounts of high quality heat (500 – 1000 °C), and produce relatively low quality heat (e.g. 100 °C) as a waste product [37, 273]. Whereas this heat is unsuitable for most industrial applications, it is still useful for residential and commercial heating applications. Therefore, many urban areas have a district heating network, i.e. a series of pipes that transports hot water from a source to consumers, who can use this surplus heat for space heating and for hot tap water. Power plants are a popular source of surplus heat.

To meet the heat demand also when there is not enough surplus heat, these district heat grids usually have an auxiliary heating system. Hereby, the scale of heat grids justifies the investment in high value heat production facilities, such as cogeneration units (i.e. CHPs, discussed later) and ground source heat pumps, and in techniques to support the use of alternative fuels, such as biogas. Note that in many heat grids, especially the smaller ones, no large source of “waste” heat is available, but the auxiliary heating is the only source of heat. District heating systems come in many sizes and different configurations [135, 285].

A.2.1.2. COGENERATION / CHP

A cogeneration plant produces multiple types of energy simultaneously. Just as in district heating, the integration of energy processes gives opportunities to improve overall efficiency. The dominant example of cogeneration is *combined heat and power* (CHP), which produces heat and electricity simultaneously. These units have a perceived high efficiency, because the heat is not considered as a waste product. However, the electric yield is modest (20 – 30% [81]) in comparison to a dedicated power plant. Therefore, the electricity production is considered as a byproduct

of the heat production; due to the low efficiency of CHPs in electric generation, running a CHP specifically for electricity production while dumping the heat is economically unsound. Most CHPs use natural gas as a fuel source. In the vision of the energy transition, CHPs can be considered as renewable generation, provided that the gas comes from a renewable source (e.g. biogas [81], or gas from a waste water treatment facility).

The energy transition asks for an electricity driven operation of CHPs. By thermal buffering, CHPs give opportunities for flexible electricity generation, which allows the heat and electricity production of the CHP to be decoupled. The heat is stored in a thermal buffer, which is usually a large hot water vessel. The buffer can subsequently satisfy the heat demand. By this, a CHP can operate in an electricity driven mode without heat dumping, as long as the heat buffer has sufficient capacity, and the time between the production and consumption of heat is not too long (i.e. days) to avoid excessive storage losses.

The heating of greenhouses is a major application of CHP in the Netherlands. In this application, not only the heat, but also the CO₂ production of the CHP process is used for fertilization. An increased CO₂ concentration lets plants grow faster. The resulting electricity sales are a major source of income for greenhouse farmers, and are often optimized for sale on the APX day-ahead electricity market [197], and are thereby a major player on the energy market. In 2013, greenhouse CHPs produced 10%, and consumed 6 – 7% of the total electricity demand in the Netherlands [336]. By this high contribution to the national energy balance, greenhouse optimization (including the CHP) is a key element of energy efficiency plans [197, 309].

CHPs are also used on a smaller scale, for block heating (i.e. a group of houses) and the heating of individual houses. These are known as miniCHP and microCHP, respectively. Small-scale CHP systems tend to have a fairly low (5 – 15%) electric efficiency [81]. Fuel cell based CHPs promise to improve this dramatically (to 60% electric and 30% heat efficiency), at the cost of a smaller thermal power, a less flexible operation, and higher investment cost [92]. The potential use of larger groups of microCHPs on the day-ahead electricity market is the central topic of recent research [19, 40, 231].

A.2.2. SMART GRIDS

The increase in dependence on electricity as an energy carrier, resulting from the energy transition, asks for improvements to electricity grids and the connected resources, such that these can supply more electricity demand, and integrate more distributed, renewable energy sources. In a business as usual scenario, this would mean capacity expansion, i.e. adding more cables, transformers, and peak generation capacity, which is very expensive.

Smart grids propose to improve the utilization of existing infrastructure and the connected resources, rather than increasing capacity, which provides an alternative to capacity expansion. The smart grid should be viewed as a paradigm for the design and operation of energy systems, rather than as a specific combination of technologies. Nevertheless, labelling the individual parts that satisfy the descrip-

tion as a smart grid (ready) solution is common in practice. While each of these parts can locally improve utilization, integration is an essential, yet often overlooked part to realize the full potential of the smart grid concept.

In this section, we first discuss the goals of a smart grid in Appendix A.2.2.1. Next, we review specific smart grid approaches. We make a distinction between “smart hardware”, utility scale smart grid techniques in Appendix A.2.2.2, and “smart coordination”, distributed smart grid techniques in Appendix A.2.2.3. The smart hardware, utility scale perspective usually come from a power engineering and industry context, and the smart coordination, distributed perspective from an academic context. Both have their own potential and shortcomings. In this thesis, we choose to focus on the smart coordination perspective with distributed techniques that control small devices; therefore, Section 2.3 continues with a discussion on the coordination of energy systems.

A.2.2.1. GOALS

The overall goal of “the” smart grid is to maximize the (social) profit of energy or to minimize the cost from energy, including the cost for the environment – i.e. to reach the goals of the energy transition in an efficient way. This is a very general goal; the use of the term smart grid emphasizes that this may be reached through unconventional means.

We present an overview of the problems that are addressed by smart grids. These goals are not always in alignment: the optimization of the grid infrastructure may conflict with the optimization of local resources. For example, heat buffering can relieve stress on the grid, but gives extra thermal losses. By cooperation, the value of the available infrastructure can be improved substantially [247, 249]. Ideally, a smart grid should balance the goals, such that it pursues the global optimal behaviour.

In the following, we outline a small, representative set of typical goals: load balancing, energy security by islanding, cooperation, and market integration.

A.2.2.1.1. Load balancing The energy transition leads to an increase in the variation in electricity supply and demand over time. By load balancing, transport losses and other inefficiencies can be decreased. Variations decrease the efficiency of the electricity supply chain for multiple reasons. Downward variations let plants run in a lower, possibly less efficient running mode. The switch between levels introduces losses as well; a gradual switch between production levels is preferred to more sudden changes.

In the grid (i.e. cables and transformers), the losses follow a quadratic relation to the current. Therefore, an electricity demand profile that is balanced over time decreases transport losses in comparison to a profile with variations. The losses are in principle symmetric, independent of the direction of flow (in an *alternating current* (AC) grid, exceptions do exist with respect to non-unity power factor loads and imbalance between phases). A common abstraction to limit losses is to impose peak demand limits, which ensure that the losses do not surpass a chosen level.

A.2.2.1.2. *Technical constraints* Smart grids are used to enforce technical constraints, such that the system can use less expensive components, or keep existing components longer in operation in spite of an unanticipated growth in supply and/or demand.

At a transmission level, the net peak demand is an important aspect of an electricity system, because it constrains the minimum necessary installed and operational generation capacity. Conversely, the lowest demand, or base load level, determines the amount of demand that can be satisfied with high efficiency base load plants (note that with the advent of the energy transition, even “base load” plants increasingly support a wide power output range nowadays). Furthermore, plants have ramping limits, which are usually better satisfied on a (near) balanced profile as opposed to a varying profile with sharp gradients. For a given energy content, a flat demand profile over time has the lowest possible peak demand and ramping requirements: any variation would increase these parameters (however, a completely flat net demand profile is usually not economically reachable, and only required at the system level).

In the grid, the peak demand on specific resources (transformers, cables, fuses) serves as an approximation of various operational limits. Due to the quadratic relation of loss to current, a peak demand reduction contributes to a reduction in losses. The voltage drop and/or rise must be limited to ensure compliance with power quality regulations (e.g. [242]). The aforementioned losses do not only waste electricity, but the accumulation of heat can also damage the resources or decrease the lifetime. Note that some resources may be overloaded for short periods of time (i.e. hours), although this comes at the cost of reducing the lifetime.

A.2.2.1.3. *Islanding / Energy security* To improve the resilience of the electricity supply, users want to be able to operate their part of the grid independent from the central infrastructure. The independent, disconnected operation of a part of the grid, a “microgrid”, is called *islanding*.⁴ This islanding requirement particularly arises in economically or life critical applications (e.g. data centers and hospitals), and in areas where grid failures are frequent enough to be a nuisance, such as in large parts of the US. For this reason, nowadays a large number of households has backup generation there, as do most businesses; an estimated 9 million islanding capable generators were operational in the US in 2011 [22]. In the Netherlands, the current high reliability of the grid infrastructure (23 minutes of unavailability per year on average [89]) gives little incentive for islanding, except for critical facilities such as hospitals. However, large outages such as the one in March 2015 in the Netherlands [316] may change this perspective.

The energy transition gives new opportunities for islanding. Rather than using a dedicated backup generator which is only used during emergencies, (renewable) distributed generation can provide this generation as well. A smart grid can optimize the use of the available generation and storage, for example to maximize the time that the islanding can be sustained, or to minimize wear on the storage resources. Furthermore, if a microgrid is set up, then a smart grid may support resource sharing.

4. Islanding also refers to the procedure itself, i.e. the physical disconnection from the grid.

Islanding is a specific case of load balancing, where a total balance between supply and demand must be reached at all times within a microgrid. Furthermore, the microgrid has to maintain system stability, and power quality. Conventional generators have inertia from rotating mass, which helps to stabilize the system. In an energy transition scenario without conventional generators, extra control may be needed to compensate for this decrease in passive stability. A smart grid may be (part of) a solution to maintain a stable electric island.

After a catastrophic system failure, i.e. a blackout, one of the challenges in power system restoration, or a black start, is that a lot of devices will turn on during the restoration effort, which synchronizes start-up effects. Furthermore, after hours or days of outage, all devices and in particular thermostatically controlled loads will be in an “empty” state, which means that all devices want to run, which is described as a cold load pickup. Whereas smart grids aim to prevent such events in the first place, smart grids can also help to stop cascading failures, and reintroduce loads in a controlled way.

- A.2.2.1.4. *Cooperation* The smart grid paradigm strives for the optimal use of energy resources. Although transport cost favours locality, the optimal allocation may often use resources that are far away, e.g. a large power station rather than a local diesel generator. At a smaller scale, it may be beneficial that the neighbour’s EV charges at a lower rate when your washing machine is running. These examples demonstrate that the ability to control and exchange energy streams, and the ability and willingness to cooperate in the use of energy resources are important for an efficient energy system.

The energy transition increases this need to share, because most RES have a variable feed-in profile and are distributed throughout the grid. A portfolio of RES can offset this variability. However, keeping many small units of different types is inefficient due to economies of scale, which favour large units and diverse groups to allow for statistical averaging. Furthermore, RES benefit from spatial diversity: PV panels and wind turbines that are close together each exhibit highly correlated behaviour, and are subject to the same local grid conditions. Similarly, households have diverse demand profiles, and may significantly improve the utilization of their energy resources by sharing.

- A.2.2.1.5. *Market integration* A complication in energy cooperation is that several resources have multiple stakeholders, each with different objectives. Therefore, a practical smart grid system needs to consider not only streams of energy, but also streams of value. Value usually translates to money, but can also describe more intangible benefits, e.g. faster EV charging, comfort, or clean air [247].

The liberalization of the energy market around the 1990s plays a crucial role in the structure and organization of the electricity grid: electricity companies are split up into generation, transmission, distribution, and retail companies [174]. The hurdle in realizing the smart grid vision is that not all energy resources are owned by the same entity, i.e. there are multiple stakeholders. The liberalization has increased vertical fragmentation: the grid, production facilities, and retail companies are no

longer owned by the same company, which gives less motivation for cross-layer optimization. However, the unbundling does decrease horizontal fragmentation, as it promotes the sharing of e.g. generation resources, and enables effective competition between retail companies. The grid itself is a natural monopoly,⁵ and is thereby exempt from competition; usually, the grid companies (i.e. *transmission system operators* (TSOs) and *distribution system operators* (DSOs)) are either state-owned or under tight government control.

The current, liberalized grid organizes the cooperation in markets, where parties can trade energy commodities. Energy trade means that the parties agree to inject and consume energy in a specific way at specific times into/from the grid, and agree on penalties for behaviour that differs from these agreements. For larger parties, this is usually a production or consumption pattern, with (often asymmetric) penalties for violations of the agreed pattern. Note that local imbalance, which is counterbalanced by synchronized imbalance elsewhere, is an intentional feature of energy trade, i.e. a transaction. Smaller parties interface to the market through a retailer. For example, an electricity retailer can ask different unit prices at different times (e.g. day-night tariffs), reward the ability to shape demand, or penalize the inability to follow a demand profile. We will address markets in some more detail in Section 2.3.1.2. Many smart grid approaches adopt the existing market structures that are already operational, but are often not adapted to a smart grid context.

In current practice, the grid operators and their infrastructure are usually not seen as a part of the market. Instead, these operators have to build infrastructure to satisfy the demands of the market, and ask regulated tariffs for use of the infrastructure. Legislation restricts foreseen undesirable transactions to keep the infrastructure affordable. In the current legal framework, a grid operator has limited freedom to participate in, and be part of the market. Therefore, their use of smart grids is restricted to techniques that improve the internal operation of the network. However, in the smart grid paradigm, the grid infrastructure is actively considered in the allocation of resources, as it can be one of the bottlenecks in the system. Hence, legal issues play a major role in smart grid developments [247], especially in non-experimental settings.

A.2.2.2. SMART HARDWARE FOR SMART GRIDS

As discussed at the beginning of this section on p. 253, there are two distinct perspectives on smart grids. The smart hardware perspective on smart grids comes from a practical setting. The energy transition forces utilities to transport more energy. The most straightforward way to be able to transport more energy is to increase capacity by reinforcement, i.e. adding more cables, and larger transformers. However, this method comes at a large cost, in particular in existing built up areas

5. Building multiple competing electricity distribution infrastructures, as e.g. has happened with the Internet (telephone, cable, fiber), is considered to be too expensive and harmful to economies of scale.

with underground cables (see e.g. [HAT:7] and references therein). Therefore, utilities look for alternative ways to increase the capacity, preferably without interfering with a business as usual operation of the grid.

Most smart hardware, utility scale smart grid techniques work at the transformer level, i.e. add a component to, or change the design of the distribution transformer or a substation. In comparison with the approach we discuss later on p. 259 (and focus on in the rest of this thesis), this is a “centralized” approach.⁶ This centralized approach has several advantages that make it popular in practice, e.g. there is limited need for cooperation, and economies of scale apply. However, a major disadvantage is the lack of (re)use of resources, which leads not only to redundancies (e.g. using a battery to charge an EV, which essentially is a moving battery), but also ignores local opportunities within the grid. For example, a central battery (e.g. for the storage of PV) can decrease transformer load, but may not reduce the load on cables in the grid, as the power still has to flow through the distribution network to access the battery.

A.2.2.2.1. Physical constraints Energy systems expect the infrastructure to deliver energy with certain properties, such that systems can rely on these properties. The infrastructure itself has similar physical constraints. These properties are usually standardized, to allow for interoperability. For example, European distribution grids have a nominal phase voltage of 230 V, with a 10% (23 V) tolerance in both directions, and a frequency of (almost) exactly 50 Hz [242]. For the electricity domain, these kind of energy distribution constraints are described as power quality (however, since the voltage and frequency are crucial for the operation of the system, these are usually treated as separate subjects).

With the introduction of DG, voltage norm violations are becoming more common. Grids need to account not only for voltage drops in the grid, but also for voltage rises. Therefore, the conventional design practice to statically configure transformers at the upper end of the legal voltage bound is no longer valid. Especially in rural areas with large amounts of PV generation, this leads to overvoltage problems [248]. Although configuring the transformer toward the lower end of the range can solve this problem, this may lead to violations at the lower bound. An *on load tap changer* (OLTC) allows the transformer to automatically reconfigure the secondary voltage while in operation. Thereby, the transformer can adapt both to periods of high feed-in and of high demand. However, a grid with an OLTC at the transformer is still limited to the rise–drop voltage bandwidth within the grid.

Transformers and cables heat up under load, which leads to wearing of the infrastructure. Therefore, to prevent excessive damage, operators set limits on the currents through the resources, which are enforced by fuses. These limits can be violated for short periods of time (e.g. a 10% overloading can be sustained for hours [157]), although this decreases the available capacity later on. The rated current takes a conservative estimate of the cooling by the environment, and of the system’s operation history. Recognizing that aging is determined by the tempera-

6. At least in a typical Dutch configuration, where a transformer services hundreds of households.

ture rather than by the load (which increases the temperature), a possible way to increase the capacity of the grid is dynamic rating, which accounts for (predictions of) the temperature of resources rather than for current flows [84, 157]. While this technique may be integrated relatively easily for a transformer, measuring the temperature of existing underground cables is difficult. Dynamic rating is already used in the transmission grid [84]. Nevertheless, limiting the load on the grid remains a good practice to reduce wear on resources.

A.2.2.2.2. Storage Centralized storage at a below-transformer level can avoid supply and demand peaks on the transformer or on specific sections of cable, and thereby avoid the need for reinforcement. In some difficult situations, storage is already the economically preferred option over conventional reinforcements [250]. Storage technologies differ by orders of magnitude in cost, economical unit size, and the ratio between energy stored and power (i.e. the speed by which the storage can be charged and discharged) [250].

A vast array of storage technologies are available on the market, and storage technologies are under active research. Options for storage during a short amount of time are electrochemical batteries [78], biogas buffers, and heat storage, among others. For storage over longer time periods, redox flow batteries and *power to gas* (P2G) are considered [86, 250].

At a transmission level, hydroelectric power is a major storage resource. Next to a variable production (which depletes the reservoir), pumped hydro uses the ability to pump back water into the reservoir to store excess production (see Appendix A.1.1.4). Note that the remote location of hydroelectric facilities in general precludes the use of this resource for local congestion management. Furthermore, *compressed air energy storage* (CAES) is considered, which promises to offer a comparable resource with different geographical and operational constraints [216].

As an alternative to operating storage itself, the grid operator may give incentives (or push for legislation) that lets others operate storage in a way that benefits the grid, e.g. with self-consumption of DG (Appendix A.2.2.3.6), or by cooperation with a storage operator [249]. However, legal barriers often prevent this type of cooperation, or lead to adverse effects [247].

A.2.2.2.3. Monitoring Insight in the actual behaviour of a resource rather than only using design rules allows a grid operator to make informed decisions on the operation and maintenance of the grid, which makes a less conservative use of resources justifiable and safe. Active monitoring of resources, e.g. by distribution automation and substation automation, allows for a dynamic control which accounts for the conditions at hand [114]. The major 2003 northeastern US black-out is attributed for a large part to a lack of situational awareness at the transmission level, which has led to the widespread introduction of *phasor measurement units* (PMUs) [243].

Smart metering, also known as *automated meter reading* (AMR) (unidirectional, “upward” communication) or *advanced metering infrastructure* (AMI) (bidirectional communication) provides measurement points throughout the grid, which may

report the power flows and voltages at each connected household, and send back e.g. real-time prices. Applications of smart metering include avoiding manual meter reading, outage detection, and detecting losses (of both a technical and non-technical nature, i.e. electricity theft) in the grid [114, 179]. Smart meters are in principle a distributed technology, but are operated in a centralized way. In the Netherlands, smart meters have a standard “P1” read-only serial port to give consumers access to measurement data at a 10 second interval [241].

A.2.2.2.4. *Transmission* Whereas most smart grid activity focuses on the periphery of the grid, the energy transition touches the transmission grid as well. In the transmission grid, “smart” is business as usual. Many smart grid innovations originate from the high voltage domain – e.g., transformer reconfiguration, fault tolerance, and heavy industry demand response are commonplace elements of a modern transmission system. Industry has developed a large body of experience with the control of these systems; therefore, we discuss the control at a transmission level in Section 2.3.1.

Current energy transition activities in the transmission domain include large capacity expansions (and their planning), the connection of energy markets, and the operation of system balancing. RES increase the need for balancing, in particular at specific moments of uncertain feed-in [123]. The capacity expansions involve increasingly high voltages, and are increasingly based on *high voltage DC* (HVDC) to reduce the losses on long distance lines.

A.2.2.3. SMART COORDINATION FOR SMART GRIDS

An alternative view on the smart grid is a smart coordination, or distributed perspective. The energy transition increasingly connects energy resources on the periphery (i.e. the bottom, low voltage edge) of the grid. From a conventional perspective, this is a problem: these resources are connected to a part of the network with limited transport capacity and relatively high losses for the amount of power and distance travelled. To connect all loads and accommodate all renewable generation, large network reinforcements would be needed.

From a transition perspective, the spread of energy resources across the grid is not a problem, but part of the solution. Local generation and consumption means that power does not have to travel all the way down the electricity infrastructure. Locality does not only reduce losses, but also avoids capacity bottlenecks in the grid, provided that the generation source and the load are close to each other. Locally consumed power does not contribute to the load on the transformer, and avoids load on cables. The large number of small units potentially provides massive redundancy, allowing security constraints to be met with ease (however, the question remains whether the devices or the *virtual power plant* (VPP) itself should be seen as a unit from a reliability perspective). With storage, even more energy can be kept local, and demand can be balanced over time. The distribution grid is no longer viewed as a network for the distribution of central production, but instead serves as an infrastructure to share energy. Consumers are no longer passive clients of an electricity retailer, but take the role of *prosumers*, who contribute to the energy supply.

Although prosumers generate energy, current prosumers do in general not behave as “proper” energy suppliers. For example, a prosumer does not have an obligation to produce energy. In general, the grid is treated as a convenient “dumping ground” for excess production, without taking into account the benefit of the supply for the system. Distributed smart grid techniques aim to improve the use of *distributed energy resources* (DER), such that their behaviour matches better with the needs of the rest of the system. In return, the system allows for a more liberal use of the available resources.

We address the distributed smart grid approach as follows. First, we present a distributed smart grid concept with an analogy to the Internet, and compare how it matches *and* how it differs. Next, we discuss a related set of concepts that also have emerged in the context of distributed smart grids. In these concepts, the coordination of DER plays a central role. Therefore, we continue in Section 2.3 with a discussion on the coordination of energy resources. As in this section, we again start from the perspective of the conventional grid, and move towards smart grids.

A.2.2.3.1. *Internet of energy* There are a number of reasons to set up an analogy between a (distributed) smart grid and an *internet of energy* (IoE); see [11, 46, 195, 340]. This analogy offers several interesting parallels. The Internet enables computers to exchange information; the IoE enables DER to efficiently exchange energy. Peer-to-peer technologies such as BitTorrent, use the end nodes of the Internet to disseminate information, exploiting multiple sources and locality to improve download speed. The concept of caching, i.e. storing information for later reuse, corresponds to storage.

However, there are also several important differences – information and energy are physically different. A major difference is that copying energy is not possible. Therefore, “caching” means that the energy can *only* be used later on – at most once. A further difference concerns losses. Energy transmission losses can be repaired by retransmission; however, this loss contributes to the consumption of energy. In an Internet context, this would correspond to a waste of processing effort and network bandwidth, but not to a waste of information. In contrast to an energy infrastructure, overloading in an Internet context does not lead to permanent damage to the infrastructure and to devices, but only to a temporary degradation in *quality of service* (QoS). Being subject to (a variant of) Moore’s law, Internet bandwidth has become virtually unlimited for all but the most demanding applications; in the energy domain, “bandwidth” (i.e. transport capacity) is a scarce resource. Therefore, the Internet is more tolerant to ad hoc capacity management.

However, energy also has several favourable properties in comparison to information. Information is volatile – fresh information is more valuable than old information. Interactive services are very sensitive to latency rather than to bandwidth. In contrast, fossil energy is very old, yet just as valuable as new energy. However, many energy carriers do have losses over time, and latency is a crucial aspect in the primary control of electricity systems. Users prefer information from specific services. Energy comes as commodities: it does not matter whether it is

produced by a large plant or by a PV panel, as long as the commodity is of the right type and of sufficient quality (while this is true from a technical perspective, consumers seem to prefer “green” and self-produced electricity). Some hybrid energy systems even allow a choice between multiple types of energy.

The Internet underlines several design patterns that are present in conventional electricity systems. For efficiency, web sites are not hosted by arbitrary PCs, but by dedicated, well-connected hosting servers (“power plants”); the current trend of server consolidation at cloud computing services pushes this concept further. Despite that vast processing capabilities are also available locally, software applications are transformed into apps, i.e. thin clients for applications that run “in the cloud”. Similarly, while it is possible to reliably generate power with a local generator, a connection to the grid is less expensive. These are examples of centralization, which are motivated by a reduction in maintenance effort. There are physical limits to centralization: locality matters for latency, available bandwidth, and reliability; for energy, locality matters for losses, transport capacity, and reliability. Consumers have connections that are aligned towards “downloading”, and share a larger connection towards (the rest of) the Internet. The dimensioning of this shared connection relies strongly on the diversity of the demand of its users.

A major lesson from this analogy is that there is a constant tension between centralization of resources on the one hand and distribution on the other, and that the outcome changes over time. The available technology, as well as opinions on the importance of different aspects, determine the outcome of the tradeoffs that are made. Therefore, a smart grid approach should balance between the use of central and local resources.

A.2.2.3.2. *Demand side management* The smart grid aims to take control of the energy streams in the grid, using distributed energy resources. However, most DER are relatively expensive in investment and operation in comparison to centralized resources, in particular in small-scale, residential configurations. As an alternative, *demand side management* (DSM) proposes to exploit the flexibility within the demand. By this, DSM transforms the demand to a distributed energy resource, which gives a promising way to provide energy flexibility without investing in “real” energy resources. Many devices with a large energy consumption have an intrinsic storage capability, which allows the load on the grid to be shifted in time. For example, a washing machine usually does not have to start immediately, but can wait for an opportune moment to run its program, provided that the user is reasonably cooperative, i.e. fills the machine as early as possible, and is patient enough not to request it to finish as soon as possible. Just as most other sources of DSM flexibility, this example relies not only on the flexibility of devices, but also on the flexibility of its users.

DSM is a concept that is older than smart grids; however, the term has acquired a new meaning in recent years to accommodate the energy transition. In the late 1970s and 1980s, the electric utilities in the US started to realize that capacity expansion is not the only possible answer to a growing peak demand. Instead, these utilities started to attack the demand side. Incentive programs and subsidies mo-

tivate consumers to buy energy efficient devices with a higher initial cost, which benefits both utilities (who have to serve a lower peak demand) and consumers (who have to buy less electricity). Furthermore, the utility can set up schemes to ask its consumers to use less electricity at peak times (usually in exchange for a reward), i.e. to shift or curtail their load. This paradigm, including both energy efficiency and load management programs, is called demand side management, and gives an alternative to capacity expansion within an *integrated resource planning* (IRP) approach⁷ [372]. In [371: p. 51] (“PG&E’s Integrated DSM model”), different “stages” of DSM are considered with an increasing impact on operations, starting with easy measures (energy conservation) and going towards more complex measures (self generation and changes in daily operation). Due to their large contribution to the peak load and their easily shiftable nature, most of these DSM programs focus on air conditioning load. Many of these programs are highly successful, and continue to contribute significantly to maintain a reasonable demand profile [346, 371]. For an international (yet mostly European) perspective on DSM, see e.g. [169]. A specific, popular type of load management is *demand response* (DR), which allows utilities to curtail loads when the reliability of supply is threatened, or to give dynamic, time variant prices to reflect the conditions on the market [4, 371]. By this, the utility can relieve stress on the grid at the point where it is most needed. If the DR event has a “pre-charging” effect or a recovery effect afterwards, then it is called load shifting; if the load does not reappear, then it is called load shedding. DR with HVACs usually exhibits both, i.e. the cooling load decreases as a result of tolerating a higher temperature, and the temperature needs to be brought down to the original range afterwards. DR events may be communicated through several means, e.g. by telephone, a smart meter or over the Internet. With *automated DR* (ADR), utilities can invoke DR resources without manual intervention, which allows for faster control and saves management effort.

In the smart grid, dynamic load management is not just an emergency response tool, but a central part of day-to-day operations. The smart grid treats flexible supply and demand resources equivalently, using whichever is most beneficial. By this, smart grid DR will be invoked far more often (i.e. possibly multiple times per day) than conventional DR (usually limited by contract to a maximum of 10 – 15 events per year). Therefore, automation is essential in this configuration. When DR becomes an everyday practice, the tolerance for a loss in comfort will decrease, which means that only unobtrusive forms of DR may be used frequently. Furthermore, as DR is used at noncritical times where supply side flexibility and storage alternatives are still available, DR competes with these flexibility providers; consequently, the value of the flexibility per invocation is much lower than in conventional, critical peak DR. Therefore, a smart grid DR operator will probably give less generous rewards to users than for conventional DR per invocation (however, the total value over the year is higher due to a far more frequent use of the service, depending on the cost of use, i.e. user reward plus DR operator mark-up).

7. An utility can choose either spend money on peak capacity, or on incentives that avoid the need for this capacity, or some of it on both.

Smart grid DR usually targets a large aggregation of devices and appliances of residential customers, with the goal to make available a large amount of flexibility at low cost. In contrast, conventional DR focuses on the “big spenders” that are active during critical peak periods as an emergency resource (i.e. to be used only a few days per year) – in particular HVAC loads. Recently, lighting has also received attention in commercial environments, due to changes in Californian legislation [53, 55]. Consumer DR almost exclusively focuses on HVAC, and on manual response to periods that are announced to have a “critical” status.

In modern use within the context of smart grids, the term DSM refers almost exclusively to load management by this form of “pervasive” ADR, usually in combination with other local DER. As a result, it would be more accurate to refer to DSM as *decentralized* (or *distributed*) energy management rather than as *demand side* (energy) management. Although the term DSM is ambiguous from a historic perspective, we adopt the modern variant of the term, and choose not to introduce a new term to refer to this concept.

A.2.2.3.3. *DSM flexibility* A smart grid system has to control distributed energy resources in a structured way. For this, it requires models of the behaviour of devices. A model can be as simple as e.g. “if I turn on a device, then the demand increases”, or as complicated as a highly detailed, cycle-level simulation model. Users are a crucial element of DSM resources. Therefore, a model of DSM resources should not only model the characteristics of the device, but also the behaviour of the user in relation to the resource. A DSM model needs to capture the aspects of the user that affect the flexibility of the device. These aspects are the tolerance of the user for control (i.e. “user flexibility”) and the way the device is used.

To determine the flexibility, a system can either make (possibly static) estimates, i.e. learn the available flexibility based on experience, or explicitly ask the user to specify the flexibility. Forms in between are possible as well. An example of the first is a thermostat, which “knows” that users tolerate a band (e.g. $\pm 0.5^\circ\text{C}$) around a set point. Whereas the set point is provided explicitly, the band is static and conservative, serving as a control dead-band rather than as a source of flexibility. In DR applications, the set point is changed to encourage or to postpone consumption, but usually the flexibility within the band is not exploited (however, the overall consumption does vary by set point). A smart thermostat may try to find the flexibility with experiments on the user: it can gradually expand the bounds until the user starts to “complain”, i.e. manually override the control. The last example already gives a partially explicit flexibility specification: users give feedback on the performance of the control at some points in time, from which the controller may infer the (limits on) flexibility.

Instead of using estimates of the flexibility of users, the system can also explicitly ask the user to specify the available flexibility, i.e. use estimates that are provided by the user. For example, the user can specify the time window within which a washing machine has to run, or give the next departure time of an EV, thereby specifying the time at which it should be charged. The explicit approach is implemented in various experimental “smart grid ready” devices and appliances,

either as an extension of the device [210: Figure 21, 334, 177: p. 14, 229, 98: p. 7], in a home energy gateway with a central control display [95, 176], or as a smartphone (web) application [94]. Advantages of this approach are that it is relatively easy to implement, that it can incorporate user information that is not represented in the system, and that the responsibility for a faulty specification can easily be assigned to the user. Disadvantages are that it requires more user effort, and that the user may provide inaccurate or overly conservative parameters, either out of convenience or anxiety. If the flexibility configuration takes too much work or intellectual effort, then users will stop providing flexibility [210: p. 86]. An energy management systems needs to strike a balance between automation and autonomy on the one hand, and end user control on the other hand [125].

Various hybrid approaches between a learning and an explicit flexibility characterization are possible. For example, a system can propose reasonable flexibility settings, speculate on flexibility that will be specified in the future (e.g. an EV that tends to come home every working day at around 18:30), or prepare for common “change of plans” scenarios (e.g. presume an earlier deadline for EVs that are often disconnected before the end of the charging window).

The concrete use of a device affects its flexibility as well. For example, the energy consumption during an EV trip affects both the minimum charge that needs to be in the battery on departure, and the quantity that needs to be recharged on return. The way of use is usually considered to be essentially inflexible: large price differences or social pressure are needed to motivate real flexible behaviour (e.g. drive less when energy is scarce). Controllers generally make conservative estimates about the way of use, e.g. always charge an EV to a (near) full level before departure, or ensure that a heat buffer can serve a shower session at any time. By this, users have come to expect a high, always-available comfort level, which is also expected from controllers that implement DSM. These expectations limit the available flexibility, and risk user complaints if the system is not used in a specific way (in turn, this causes system problems, i.e. an unwillingness to participate in DSM, or demands for higher rewards). Consequently, DSM systems usually model the comfort limits as hard constraints.

The difficulties of the DSM flexibility problem lay not only at the user side, but also with the configuration of the demand, which promotes conservative control. The modern devices of the energy transition use slow, buffered processes that do not always allow for an efficient, fast, just-in-time response to demand to the extent that we want to. For example, when a user decides to leave immediately with an EV, then it is too late to respond by increasing the charge speed. However, a PHEV would be able to make up for the difference by driving on fossil fuel, although at a higher cost (clearly, the extra cost should be socialized). Similarly, a heat pump with a high power heating rod can make up for insufficient heat storage. Note that whereas the use of fuel in a PHEV decreases electric load, the response by a heating rod gives a significant increase in electric load.

Summarizing, a load with DSM is not only a DER that provides flexibility, but also a source of uncertainty in the operation of a smart grid. The uncertainty concerns both the tolerance of the user and the way of use. Whereas the shape

of the demand profile is flexible, the demand volume is fairly inelastic. However, some demand may be shifted between energy domains. The system configuration of the loads that are subject to DSM determines the usable flexibility, as well as the behaviour under both normal and exceptional circumstances.

- A.2.2.3.4. *Virtual power plant* An aggregation of a large number of DGs can emulate the behaviour of a large power plant, hereby creating a *virtual power plant* (VPP). There are various implementation variants of VPPs. The most popular variant uses a cluster of microCHPs. TRIANA, the coordination system that is central to this work, originally aimed to form clusters of microCHPs and steer them as a VPP, with support for optional islanding operation [19, 40, 231]. Other variants use backup generators or batteries (either by synchronous generation or by voluntary islanding) [100], DSM (i.e. emulate “production” by decreasing load) [99], or a combined portfolio of DER, including both production and flexible demand resources [120].

VPPs have many similarities to DSM. Both rely on an aggregation of DER. However, the focus of a VPP is on the flexibility of the production resources, and for DSM it is on the flexibility in the demand. Also, a VPP generally aims at scattered resources, whereas DSM typically considers resources that are clustered in one distribution grid. For a VPP, locality is a problem rather than a solution.

Similar to DSM, user behaviour has an impact on the operation of many VPP configurations. For example, since the electricity and heat production of a microCHP are coupled, the user’s heat demand determines the electricity production, unless heat dumping is possible and economically acceptable (i.e. gas is cheap).

- A.2.2.3.5. *Dynamic curtailment* The production peaks of RES may cause significant stress on the distribution grid, and increase the risk for voltage and frequency standard violations by uncontrolled feed-in. To mitigate these risks, modern DGs sense the voltage and frequency on the grid connection, and decrease production when the limits are (almost) violated. The same approach can also be applied to some loads [147]. However, as we discuss in Section 2.3.1.4, these mechanisms may give unintended dynamic behaviours (and have given these behaviours in practice). In some countries, production curtailment is mandatory for new PV installations. For example, Germany requires consumers to either curtail the net feed-in to 70% of the PV peak panel capacity, or install a remote controlled curtailment interface [224]. Similarly, grid operators can ask wind generators to curtail their production, in exchange for the market value of the curtailed production [247]. Curtailment can avoid the cost to cope with worst case feed-in scenarios; however, curtailment scenarios should be relatively rare to avoid an excessive loss in production (e.g. below 5% [224]).

Most RES have (almost) zero (or slightly positive [153]) marginal cost. Therefore, throwing away renewable energy is rational when the cost for transport and/or storage exceeds the economic value of the demand. To avoid “losing” excess RES production, people find low value applications for electricity, e.g. auxiliary resistance heating where gas heating is available (see Appendix A.2.1).

FIGURE A.3: The effect of 10 kWh of local storage on the electric demand profile from Figure A.2, according to a control policy for optimizing self-consumption (i.e. a greedy policy; hatched) and peak shaving (dotted).

A.2.2.3.6. *Local storage* Many countries have different tariffs for net demand and for net feed-in: users pay more to buy electricity than they receive for selling electricity to the grid (i.e. the retailer; see Figure A.1). By this difference, consumers have an incentive for self-consumption of local generation, in addition to the “threat” of RES curtailment. Local storage can increase the self-consumption by shifting energy from periods of excess generation to periods of net demand. Self-consumption reduces the use of the grid, and should thereby reduce the impact of DG on the electricity system (we will later, in Figure A.3, show how self-consumption may negatively affect the electricity system in practice).

In some countries, the incentive is large enough to economically justify battery storage systems: in Germany, the tariff difference is €0.14/kWh as of 2014 (see Figure A.1). Therefore, combined PV and storage systems are appearing on the market, e.g. the Nedap PowerRouter [239], and recently the Tesla Powerwall [317]. Although the number of installed PV systems with storage is still fairly small (15000 in Germany in late 2014 [44]), it is expected that this number will grow rapidly as storage solution costs continue to fall and the tariff gap increases further [107]. Several of these PV storage systems also offer support for islanding operation, using PV and the energy that is available in the battery to satisfy the demand in the event of an outage.

Whereas the intent of self-consumption is to reduce the impact of DG on the electricity system, it may in practice fail to relieve the grid at times when the stress is highest, or even increase system stress at specific times [155]. Consider Figure A.3, which shows the demand profile from Figure A.2, but now with 10 kWh of battery storage (in this example, we assume an ideal efficiency of 100%). In practice, battery systems operate according to a self-consumption maximization objective [313], which should in principle give the highest profit for the owner. We label this as a greedy control policy: it tries to charge and discharge the battery as soon as possible, to ensure the highest possible self-consumption. The hatched pattern in Figure A.3 (labelled “greedy”) gives the net demand according to this control

policy. As the grid is used only for half the width of the supply and demand peaks, the resistance losses in this example are approximately halved. However, the storage is saturated well before the end of the feed-in period, at which point charging stops. The storage barely reduces the height of the feed-in peak, and misses the top of the demand peak. Furthermore, if many PV storage systems use this policy, and the ratio between generation and storage capacity is similar, then the storage saturation events synchronize (contributing to this effect, the fill rate is highest during peak periods, and it is strongest on the days with the highest feed-in and demand, respectively), which leads to steep ramping events. The March 2015 partial solar eclipse in Europe has demonstrated that the synchronized fast ramping of PV gives a legitimate source of concern for system stability [104] (however, by appropriate preparation of flexible generation resources, the grid delivered normal service during the event, in spite of a 7 GW ramp-down and 13 GW ramp-up of PV feed-in in Germany within 3 hours [77]).

While one could argue that the storage is not large enough, a user would benefit more from installing even more PV generation capacity rather than helping the grid operator, provided that any roof space is still available, i.e. the utilization of the storage is not high enough to justify installing capacity for a complete peak coverage.

Instead of adding 2–3 times as much storage to solve the problem, a more intelligent approach yields similar results. The use of the storage may be optimized for peak shaving, as shown by the dotted pattern in Figure A.3. By this, the load on the grid is decreased substantially: in comparison to the greedy policy, the feed-in peak is 3 times as low, and the demand peak is 2.3 times as low. In contrast to the “vertically” clipped load profile (i.e. clipped in time), a “horizontally” clipped load profile (clipped in power) gives a consistent load reduction. This decreases the ramping load on the system (although there is still a substantial load reversal at 15:00, which is not considered in the objective), and gives less resistance losses (58% lower in this example). In this example, the greedy policy and the peak shaving policy give the same profit for the consumer under current *feed-in tariff* (FiT) regulations. However, the greedy policy avoids any risk of losing an opportunity to shift PV production. Also, the peak shaving policy can reduce the available charge in the battery in the event of an outage, which reduces the time that the system can sustain islanding (however, in the case of an outage in the evening, more charge may be left). Considering the lower feed-in around 13:15, note that it can sometimes be rational to discharge the battery during periods of feed-in, in order to make room for charging later on (however, discharging the battery during periods of net feed-in is not allowed in many jurisdictions to avoid accounting issues).

The peak shaving optimized use of PV battery storage clearly relies on predictions of both the supply and demand volume over time, as the controller needs to know at which “horizontal levels” it should clip the power profile (the example uses a clairvoyant prediction). The predictions may be fairly inaccurate: even with simple, persistence model forecasts and a simple adaptive control strategy, decent results have been reported in practice [313]. The controller has to balance between overestimating and underestimating the clipping levels: a too low feed-in clipping level exhausts the storage too early, giving a similar (but less severe)

feed-in peak as the greedy policy near the end of the feed-in period; however, a too high feed-in clipping level risks not filling the storage completely, which reduces self-consumption and the potential to reduce the demand peak. In the aspect of flexibility management, PV battery storage scheduling corresponds to the more general challenge of DSM flexibility management, except that a battery behaves more regularly and predictably than other DSM resources.

A.2.2.3.7. *Flexibility resources of DSM* The previous items have presented a general overview of the flexibility characteristics of DSM. In the following, we elaborate on the properties of the specific device classes that are considered for demand side management.

» *Thermal systems*, i.e. heat pumps (see Appendix A.1.2.1), microCHPs and water heaters, often allow buffering in the thermal domain: buildings heat up and cool down slowly (taking hours up to days, depending on the insulation), which means that the power consumption can take place in a wide range of time. Similarly, a heat buffer can store hot water for later use. Thermal buffers are in general constrained by temperature bounds (set points), the fill level, or more generally a *state of charge* (SoC) (0 – 100%) which describes the available flexibility. The (usable) SoC range may also be described as the energy capacity of the buffer. In a system with both heating and cooling, the SoC is ambiguous, as the concept depends on whether the system aims for heating or cooling; to be even more general, the “level” is also referred to as a state variable. Depending on the configuration of the heating system, often a distinction needs to be made between a mode for space heating and a mode for hot tap water, and these may be buffered separately (with separate state). Thermal buffers have losses over time, which gives a tradeoff between decreasing energy consumption and decreasing energy cost (by consuming more power at a less expensive time). Furthermore, many heating systems have an auxiliary backup heater to meet demand instantaneously at a high cost, e.g. a high power resistance heating element for a heat pump, or a gas burner for a microCHP.

The heat demand describes the external change or disturbance of the state, including e.g. ventilation, hot water demand, gains from solar irradiance (negative demand), which are expressed in thermal power or in change to the state variable. The heat demand can be deducted from measurements on the state variable or estimated with flow sensor measurements, depending on the type of demand. The heat demand may be predicted with a forecast of thermal demand [19: p. 37] or use a conservative estimate based on domain knowledge (“the buffer should be replenished during the night”) [189].

The heat (or cold) production gives the controlled change to the state, according to the properties and operating modes of the connected heating elements. Most low-cost buffered residential heaters only have “off”, “on”, and “on with auxiliary heating” modes, which give a known power consumption (or production in the case of a CHP) and heat production for the buffer. Some heaters allow for a variable, modulating output (and input) within a defined range; the efficiency (ratio between input and output power) may depend on the chosen modulation level.

Mode switches contribute to the wear of the device and should therefore be used with moderation; note that this again gives a tradeoff between optimizing for (the lifetime of) the device itself, and for the power consumption behaviour. The device may behave differently immediately after a mode switch, e.g. due to start-up effects (e.g. a CHP first consumes electricity, and then gradually increases power output) and shutdown effects (thermal inertia in production) [366]. Some modes may have wearing costs as well, or other costs that are not modelled as energy flows (e.g. oil consumption, and typically also gas consumption).

- » *Appliances (white goods)* with flexibility include washing machines, dryers, combined appliances, and dishwashers. These appliances often have a fair amount of flexibility: the specific time during the day when an appliance runs is not relevant, as long as the appliance task has been prepared (i.e. loaded), and the task finishes before a certain time (some tasks have extra constraints on the start time, e.g. to ensure that clothing does not stay wet for too long). After the start, a program gives a 1 – 2 hour period with extra demand with demand peaks of e.g. 2 kW from motor and resistance heating load.

Many existing appliances already have a manual delayed start function to let users exploit day–night tariffs; the flexibility of currently available smart appliances amounts to a configurable delayed start. DSM assumes that the minimum start time and deadline are given, and that the program length and demand profile are (approximately) known. The program is chosen by the user. In return, the DSM controller gives the desired start time or a start control signal. Whereas there are opportunities for short interruptions within the program (see e.g. the Grid Friendly Appliance project [147]), these are usually not considered for efficiency reasons (the rotation stops, water cools down, etc.), or restricted to defined points in the program where the impact of a delay is known to be small, e.g. between the washing and drying program of a combined washing and drying appliance.

Refrigerators and freezers also have flexibility, yet behave as thermal systems. Refrigerators have a quite narrow operating range (4 – 7 °C) to respectively prevent ice forming and decay of the contained goods. Freezers usually only have a defined upper bound (–18 °C). However, as the temperature difference with the environment increases, leakage increases and the efficiency of the heat pump decreases.

- » *Batteries* allow the storage of electricity, yet are conceptually very similar to thermal systems. A battery has a SoC, multiple operation modes (idle, charging and discharging), and losses over time. In contrast to a thermal buffer, a battery does generally not have a single tightly coupled demand, and modulating operating modes are common. The charge and discharge modes both have a maximum power; some batteries also have a minimum charge and discharge power, or discrete power levels. An efficiency factor determines the difference between the charged and discharged energy, which may depend on the SoC and the charge/discharge power (usually, fast charging/discharging is less efficient).

Batteries prefer to stay within a certain SoC range to avoid excessive wear; the specific range depends the battery technology at hand (e.g. lead-acid or lithium-ion). The number of switches between a charging and discharging (without con-

sidering the idle mode), or cycles, is a coarse indicator of the lifetime; however, the sensitivity highly depends on the battery technology and e.g. the SoC, temperature, and used charge/discharge speed [78]. Consequently, excessive switching should be avoided.

For legal, safety and/or configuration reasons, there may be limits in the use of the battery that are tied to the aggregate household electricity consumption. For example, in Germany, feed-in of battery power on the grid is forbidden (if there is net feed-in, then the battery may not contribute to it), which limits the maximum discharge power [155]. Also, a battery-led islanding operation needs to satisfy all residual demand with the battery, and is thereby bound in behaviour.

- » *Electric vehicles* are comparable to batteries, but arrive and subsequently leave from a household (see Appendix A.1.2.2). The time of leave gives the (absolute) charging deadline. The arrival time and charging deadline usually occur at predictable or specified times; in a residential context, most arrival times are in the early evening and the charging deadlines are in the morning. Driving consumes electricity from the buffer during the time that it is not connected; the amount of consumption may be estimated or specified by the EV, generally at the time of arrival. Some chargers do not give information on the SoC at all, but only stop charging gradually once the process is completed. EVs often have a minimum charge level (e.g. 6 A), and many discrete charging levels (e.g. in steps of 1 A up to 16 A); switching between these charge levels is possible. At the charging deadline, the battery needs a minimum amount of charge, which is often set equal to the maximum capacity as a conservative estimate. Most current EVs can only charge from the grid; *vehicle to grid* (V2G) describes that an EV can discharge on the grid or within the household (one of the factors that inhibits V2G is the increase in battery cycles, which decreases the lifetime of the EV battery). Depending on the context, EVs may also charge elsewhere (at a charging station or at work), but users are not considered to be flexible in this regard; in this context, the typical arrival times and deadlines differ. Furthermore, PHEVs can fulfil the driving energy demand from an alternative source, although at higher cost; a (uncommon) variant of this allows a PHEV to use the auxiliary source as a V2G generator while it is connected [183]. Given the fast introduction rate, large energy capacity, and relative non-intrusiveness of control, EVs promise to become one of the primary sources of DSM flexibility, mostly to prevent the reinforcement of the evening demand peak by the EVs themselves.

- A.2.2.3.8. *DSM demonstration projects* Over the last years, various groups have experimented with DSM in field tests across the world. We provide a few examples of smart grid projects where DSM is the core element.

In the US, major pioneering field tests are the Olympic Peninsula project [146] and the Grid Friendly Appliance project [147], both of which were completed in 2007 by PNNL. The Olympic Peninsula project controlled the electric water and space heating of 112 households, 150 kW of municipal water pumps, and two islanding diesel generators. The Grid Friendly Appliance project experimented with

50 water heater and 150 clothes dryers with a frequency sensitive load shedding controller. Both of these projects were highly successful, and provide extensive information on the observed behaviour and the opinions of involved parties in the reports.

In the Netherlands, PowerMatching City [33] demonstrates 25 households with various smart energy technologies, including microCHPs, heat pumps, smart washing machines and dishwashers, freezers, EVs, and a gas turbine. These devices are organized in a cluster, which is controlled by the PowerMatcher coordination mechanism (we discuss PowerMatcher later in Section 2.3.2.1.5). In a follow-up demonstration experiment, the same technology has been applied on a case with 288 heat pumps [3].

In Belgium, the LINEAR project gives a large scale demonstration of DSM, using 445 time shiftable appliances and various other controllable devices [210]. The report elaborates on the flexibility that the participants offer on week days and weekend days, and on seasonal influences (users appear to expose more flexibility in winter). The type of incentive and the user convenience of the flexibility configuration interface affect the response fatigue: for example, an overly complex EV interface led to a low participation. Furthermore, technical issues prove to be a major source of failure, in particular with respect to reliable in-home communication [210: p. 86].

In Denmark, *Styr Din VarmePumpe* controls 300 residential heat pumps [171]. The measurement data shows that heat pumps offer less flexibility both in summer (not enough heat demand) and in winter (full load heat demand). Also, various practical problems with temperature measurements and accounting are noted. The report gives an extensive overview of the used components and system configuration, and the efficiency of the connected heat pumps.

TRIANA (the approach that we develop further in this thesis) is demonstrated in two field tests in Germany [362]. In these field tests, TRIANA operates as the home energy controller as a client of the Smart Operator from RWE. The first field test considers a single kindergarten building with a heat pump, a smart washing machine, and a large battery [279]. The second field test uses the same system, but considers 125 households, 25 of which have various smart appliances, 10 have storage heaters, and 8 have electric storage water heaters [278]. The results of these field tests are pending.

FLEX STREET

Abstract

The comparison of DSM coordination methods asks for a common simulation scenario. The Flex Street scenario models a future Dutch neighbourhood of 400 households with various flexible devices, and aims to provide such a common scenario that can be straightforwardly ported to different simulation environments. We have contributed to the development of this scenario, and a comparative evaluation study using this scenario. This appendix describes the scenario and reports on the results of this study. Furthermore, it reports on the results of an economic study based on a variant of the Flex Street scenario. Finally, we present a mathematical model for the scenario, and determine a lower bound model for the scenario that can be evaluated efficiently. We find that Flex Street provides a flexible and convenient basis for the evaluation of DSM methods.

description 275 · model 282 · simulations 288 · case study 293 · aggregate model 296

• • • • •

B.1. INTRODUCTION

In general, it is only possible to compare DSM approaches based on a common scenario. Therefore, we have contributed to the development of Flex Street, a framework to set up DSM evaluation scenarios that allows for the comparison of simulation results between different DSM implementations. Appendix B.1, Appendix B.2 and Appendix B.4 are largely based on [HT:8].

The most common evaluation methods for individual (tertiary) control systems are case study simulations and field trials. Both of these methods are usually defined in the context of microgrids. The performance indicators used in these studies vary considerably, or, in some studies, are absent completely. Furthermore,

↪ Parts of this chapter have been published in:

- [HT:8] CLAESSEN Felix N, CLAESSENS BJ, HOMMELBERG MPF, MOLDERINK A, BAKKER V, TOERSCHÉ HA & van den BROEK MA. “Comparative analysis of tertiary control systems for smart grids using the Flex Street model”. In *Renewable energy* 69, Elsevier, SEP 2014, pp. 260–270. ISSN 0960-1481. DOI 10.1016/j.renene.2014.03.037.
- [59] CLAESSEN Felix N. “Smart grid control: an analysis of control systems within a Dutch residential microgrid incorporating decentralised renewable energy resources”. MSc thesis. Utrecht University, DEC 2011.
- [205] LAZOPOULOU Marilena. “Pricing schemes for automated demand response programmes: a quantitative analysis and comparison of pricing schemes with a smart grid control system”. MSc thesis. Utrecht University, AUG 2014.

several studies exist that provide a comparison of different control systems, all of which are based on case study simulations. In general, three different analysis methods are used for these comparisons:

1. independent simulations of systems operating on different cases, which yields a qualitative comparison,
2. simulations operating on equivalent cases, which gives a quantitative comparison, and
3. co-simulations of control systems within the same case, which enables a quantitative assessment of interoperability, competition and emergent properties.

The first two methods are mainly used to evaluate microgrid control, while the third method is used to evaluate VPP control. Although evaluation studies have made attempts to provide standardized cases, they either show a limited scope (i.e. a small or uniform device population is used) or have not actually been subjected to multiple control system architectures.

Flex Street has been set up to support the second analysis method, whereby it explicitly aims to facilitate the comparison of independently developed control systems. Furthermore, the set-up aims to resemble a realistic setting for the operation of smart grid control systems, describing a large and diverse device population.

The proposed concept consists of three steps. In the first step, the Flex Street model generates instances of a residential microgrid. In the second step, a case is made by assigning an objective to these microgrids, and defining *key performance indicators* (KPIs) for the control systems. In the final step, different control systems are simulated in a case study, and the output of the simulations is analyzed using the KPIs. A clear separation between the set-up of a case and the simulations of developed control systems has two benefits: it enables the use of pre-existing simulation environments (simulators) owned by participating developers, and it facilitates the creation of standardized cases for comparison studies.

Flex Street supports the following residential flexible devices:

- Uncontrollable loads
 - Inflexible electricity demand
 - PV feed-in
- Block shiftable loads
 - Washing/drying machines
 - Dishwashers
- Continuous shiftable loads
 - EVs
- Battery storage
- Heating
 - Heat pumps
 - Auxiliary resistance heating rod
 - Heat buffers

B.2. CASE DESCRIPTION

scenario	heat pump	washer/dryer	dishwasher	PHEV	battery	PV
Reference	0%	0%	0%	0%	0%	0%
Pessimistic	0%	20%	20%	10%	0%	5%
Moderate	50%	60%	60%	50%	5%	15%
Optimistic	100%	100%	100%	90%	20%	30%

TABLE B.1: Penetration of controllable devices in Flex Street households.

type	rated power	capacity	efficiency	arrival	prog. len.	deadline
heat pump	0 ... 2 kW _{el}		500 % (COP = 5)			
aux. heating rod	0 ... ∞ kW _{el}		100 %			
thermal storage	0 ... ∞ kW _{th}	8 kWh _{th}	99.95 %/15 min			
washer/dryer	926 ± 212 W	2.8 ± 0.6 kWh		$\mathcal{U}(0:00, 23:59)$	3 h	+24 h
dishwasher	275 ± 51 W	0.6 ± 0.1 kWh		8:00 ± 30 min (1/4)	2 h	+ 4 h
	...	...		13:00 ± 30 min (1/4)	2 h	+ 4 h
	...	...		18:00 ± 30 min (1/2)	2 h	+13 h
PHEV	0 ... 3.7 kW	3 kWh (1/3)		17:30 ± 1 h	≥ 49 min	7:00 ± 1 h
	...	6 kWh (1/3)		...	≥ 97 min	...
	...	12 kWh (1/3)		...	> 3 h	...
battery	-3.4 ... 3.4 kW	8.5 kWh	90 % (input)			

TABLE B.2: Parameters of controllable devices in Flex Street.

B.2. CASE DESCRIPTION

The developed Flex Street model represents a neighbourhood microgrid of 400 houses that is connected to the main gas and electricity grid. The houses have a selection of DER, storage options and controllable loads. We illustrate the used house model (for a house with all possible devices) in Figure B.1. The submodels that are used for all the devices are described later. The models are derived in such a way that portability between different DSM approaches is favoured over physical accuracy, i.e. we aim to make a comparison between DSM approaches and not to make a completely accurate description of the behaviour of a neighbourhood. The majority of devices is modelled in a bottom-up approach to create flexibility in case design. Flex Street currently describes the demand and supply patterns (electricity and heat) of all devices within the neighbourhood for one year, and is defined with a time granularity of 15 min. Devices are modelled with prognostic consumption/production data and usage statistics that describe the residential use in 2050 for a neighbourhood with terraced houses in the Netherlands.

Different versions of Flex Street can be generated by defining different device populations. We have three scenarios for renewable and controllable devices, which describe different penetration rates of controllable devices in Dutch terraced houses in 2050, as well as a reference scenario without controllable devices. The device penetrations rates in these scenarios are listed in Table B.1. Penetration rates were

(a) House cost control model (Section 2.4.2.1).

(b) Corresponding model in Section 3.2.

FIGURE B.1: Flex Street model of house with all controllable devices.

separated into pessimistic, moderate and optimistic rates. A fourth scenario assuming zero penetration rates was included as a reference. Batteries were allocated only to houses with PV units, and an uncontrollable gas-fired heating system was assumed for houses without an electric heating system. Fixed penetration rates were used for other devices: in all scenarios, all households have uncontrollable electricity demand, heating demand, and domestic heat demand from showers, bathroom taps and kitchen taps. It was assumed that 36% of the households has domestic heat demand from a bath. In the following, if we do not mention the scenario, then we refer to the Moderate scenario.

B.2.1. CONTROLLABLE DEVICES

The characteristics of the controllable devices are given in Table B.2. The parameters of the storage devices and DER are identical for each house. Battery efficiency is defined as round-trip efficiency, whereas the efficiency of thermal storage depends on a relative loss over time. Heat pumps come installed with a much less efficient auxiliary backup resistance heating rod, which should in general only be used during periods of very high heat demand that can not be covered by the heat pump, and for PV feed-in peaks that can no longer be stored in the battery or used for other devices.

Several types of controllable devices were modelled:

- Time shiftable appliances, i.e. washer/dryers and dishwashers, were given a constant demand profile. The rated power of these devices is described by a normal distribution, based on which a unique profile is generated for each house [59]. Control systems may shift start times within a limited time period.
 - Washer/dryers are used on 4, 6 or 7 days in the week depending on the number of occupants (2, 3, or 4 persons, respectively) [34]. Start times of washing cycles are derived from probability density functions, and are thereby of a stochastic nature. A uniform distribution over the day was used for washer/dryers. Washer/dryers are non-preemptible.
 - Dishwashers are turned on either in the morning (probability of 25%), noon (25%), or evening (50%) [34]. Dishwashers are used on 7 days in the week. Dishwasher programs are interruptible after 1 h and 1.5 h. The dishwashers within each of the three start time blocks all use the same relative deadline.
- PHEVs are fully preemptible. Where the time shiftable appliances have only a shiftable block profile, a PHEV can be controlled freely within a range of 0 W to a maximum power value. PHEVs only consume electricity, and do not support V2G feed-in. Three types of PHEVs were modelled with an equal market share and a different effective battery capacity [236]. It was assumed that the cars have to be fully recharged every day. Start times and end times of charging were described by normal distributions around 17:30 and 7:00, respectively [63].

B.2.2. UNCONTROLLABLE DEVICES

Flex Street considers a significant part of the household load, i.e. all devices that have not been modelled explicitly (e.g. freezers), to be uncontrollable. These uncontrollable loads are represented by an aggregate demand profile that lumps the uncontrollable load of a household. Datasets for uncontrollable electricity demand were obtained from measurements of Liander (a major Dutch DSO), and have been used to generate individual profiles for each house. These house profiles were also influenced by the number of residents.

A different type of “uncontrollable” load in Flex Street is the heat demand, although heat can first be stored in a buffer. The model does not differentiate between the quality of heat: the heating demand and domestic heat demand are connected to the same source. Datasets for the heating heat demand and domestic heat demand were synthesized using a heating demand model [339] and a tap water model (Appendix B.2.3), respectively. This has been realized in such a way that each house has an individual profile. For reference, we give the complete heat demand for 10 households in the first three weeks of the dataset in Figure B.2.

PV units were modelled separately as uncontrollable generators. The supply profiles of these PV units were obtained from irradiation measurements of VITO. All households that have a PV unit use the same supply profile.

B.2.3. HOT TAP WATER MODEL

The domestic heat demand profiles for each house were determined by a hot tap water submodel, using statistics on household occupancy, diurnal patterns and water use. Statistics on household occupancy for two-or-more-person households in 2050 were used to assign a number of residents to each household [85]. For each resident, we composed a unique consumption profile based on residential diurnal patterns of business travellers, adapted from [34]. Probability distributions of hot water use events were used to randomly determine in the start time of water use events. Any event is assumed to occur at the start of a 15 min time interval. The average heat demand for hot water use events is $29 \pm 2 \text{ Wh}_{\text{th}}/\ell$ of tap water, based on 100% penetration of waste water heat recovery in 2050 [59]. Using these statistics, the model composes an individual domestic heat demand profile for the entire duration of the simulation for each household.

B.2.4. UNCERTAINTY

Within the analysis and in the experiments of Appendix B.4, we assume that the control systems that manage Flex Street use the same predictions for the uncontrollable demand and supply, as well as for the controllable loads. We have created predictions with a forecast period of one day for each profile. The precision of the predictions decreases for periods further into the future. This was modelled by moving the predicted demand $\tilde{\mathbf{d}}$ away linearly from the realized values $\mathbf{d}$ for time intervals $t' \in \{t, \dots, \min(t + n_h - 1, n_t)\}$, where n_h gives the prediction horizon (96 time intervals, i.e. one day) and n_t the total simulation length (one year).

B.2. CASE DESCRIPTION

FIGURE B.2: Heat demand in Flex Street, first 21 days (in January) and 10 households.

That is, from the real demand value $d(t')$ and the forecasted demand value $\tilde{d}(t')$, $d_t(t')$ derives the demand predicted at time t for time t' :

$$d_t(t') = d(t') \cdot \left(1 - \frac{t' - t}{n_h}\right) + \tilde{d}(t') \cdot \left(\frac{t' - t}{n_h}\right). \quad (\text{B.1})$$

The values $\tilde{\mathbf{d}}$ were generated in advance by imposing a stochastic variation (“noise”) on $\mathbf{d}$. The choice of the variation values is based on the analysis of the outcomes of simple forecasting methods on the existing data (day-ahead forecasting and week-ahead forecasting, e.g. [19: p. 53]). Only the heating heat demand has been subjected to this uncertainty; the domestic (hot water) heat demand is considered to be deterministic. The *mean absolute percentage error* (MAPE) for day-ahead predictions in 15 min intervals is 18% for the heating heat demand, 40% for the uncontrollable load and 46% for the PV feed-in. To avoid division-by-zero problems on days with zero demand, the MAPE is defined in terms of the annual average demand.

In Chapter 4, we consider an alternative uncertainty model. Here, we take the real demand of earlier days as a forecast for the day at hand. An advantage of this method is that this demand can be determined in practice, and also there is no direct dependency between the forecast information and the actual demand at hand, which improves the confidence in the simulation results. A disadvantage is that the forecast that we obtain in this way is often of poor quality, and that we have little control over the forecast quality.

B.2.5. OBJECTIVE

Originally, the true objective of the case was defined informally; the journal paper [T:8] makes some assumptions on the costs caused by the outcome of the simulations (a posteriori evaluation).

In principle, Flex Street defines an extensive objective model as follows. However, typically more simple objectives are used, which e.g. evaluate (combinations of) only the peak demand, market electricity prices, or the quadratic distance from a demand of 0.

A considerable amount of diversity is generated within the collection of houses in Flex Street, featuring different demand profiles of both uncontrollable and controllable loads for each house. This set up is convenient when studying different control system objectives, as these can take into account the individual consumption patterns of households. For example, one may consider to reward a household for the amount of flexibility it has contributed to the microgrid of the neighbourhood.

B.2.5.1. PEAK SHAVING

Peak shaving is the objective in all three cases. For an electricity supplier, minimizing peaks in demand is a strong incentive to set up a microgrid. Supplying electricity at a more constant rate allows a retailer to buy from base load power plants instead of peak load power plants, which might lower the cost of electricity. Additionally, peak-shaving reduces the required capacity of the connection to the main grid, which can lead to investment savings.

To assess how well each control system has achieved the given objective, we define KPIs. The peak shaving is evaluated by the *relative peak reduction* (RPR), which is defined as the difference in peak power consumption between the reference uncoordinated control method and the control method at hand, divided by the difference between the peak power consumption and the average power consumption of the uncoordinated control method. That is, if the control method reaches a flat profile equal to the average power profile, then we have an RPR of 100%. To avoid a disproportionate impact from incidental overload conditions, peak power consumption is here defined as the capacity that covers the load of the microgrid for 99.73% of the time, corresponding to a coverage of 3 standard deviations from the average power if the load distribution would have been a normal distribution.

B.2.5.2. TRANSPORT LOSSES

First, indicators were defined to measure savings and abated emissions due to a reduction of transport losses. The required spatial structure of transport has currently not been modelled in Flex Street. Instead, a simple structure of the network has been assumed. Losses within the MV distribution network have been calculated under the assumption that the neighbourhood microgrid is connected to a high voltage network by its own power line through the MV grid. The LV transport losses for the households are computed as if each house is connected with a separate cable of equal length to the transformer. The LV and MV line losses increase quadratically with power.

B.2.5.3. DEVICE LOSSES

Second, indicators were defined to determine the *seasonal performance factor* (SPF) of heating systems and the *daily battery use* (DBU), both of which are unitless. The SPF is the average conversion efficiency of power to heat over the entire year, i.e. a variant of COP that compensates for the actual time of use of a heat pump. In the case of Flex Street, it accounts for the use of the heating rod and the losses in the buffer. The COP of the heat pump determines the maximum SPF. The DBU is the average amount of power flowing through the battery, relative to the battery's capacity. For example, a battery with a DBU of 200% has two round-trips per day on average.

B.2.5.4. INVESTMENT COSTS

Third, indicators were defined to estimate investment savings in the LV network, including abated costs for upgrading transformer capacity, cables and ditches. We modelled these costs as a function of the peak power, which corresponds to the neighbourhood's design value, i.e. the maximum average power flow in kVA per house, based on [MAT:7]. Costs of transformers, cables and ditches generally increase nonlinearly with the design value, and are typically described by jump functions that represent the discrete increase of costs due to adding components and rearranging the network. For a growing number of households and new areas (e.g. for

a complete region or country), such jumps level out and investment costs can be appropriately described by a smooth function. For a relatively small amount of households such as in the present study this model can only provide a rough estimate. To model the effect of marginal cost decrease, we used a power law for each component, with coefficients based on data from [MAT:7] and [76]. In our results, we present the investment savings per year, assuming an estimated industry standard discount rate of 6% and an equipment lifetime of 40 years [322].

Together, these indicators describe a smart grid control system's economic and environmental performance as well as on energy efficiency. On the basis of these indicators, we may compare and evaluate the control systems.

B.3. MATHEMATICAL MODEL

B.3.1. INTRODUCTION

To define the Flex Street case more clearly, we define a mathematical model for the devices in this case. The description uses the notation of Section 3.2. Devices in Flex Street do not have internal costs, but some devices have losses that may lead to extra electricity costs. The case describes user comfort requirements (e.g. when a PHEV should be done charging, and when sufficient heat should be available) with hard constraints.

B.3.2. DEVICE MODELS

In the following, we present a more formal description of the devices in Flex Street. The model is defined based on a discretization of time into intervals $\mathcal{T} = \{1, \dots, n_t\}$ which all have a fixed time interval length $\tau = 15$ min. Note that this can straightforwardly be extended to account for a nonuniform time interval length. As the case considers the duration of a year, we have $4 \cdot 24 \cdot 365 = 35040$ time intervals in total. Although we present the device model as a deterministic model and often use it as such, the model is in principle subject to the uncertainty as has been described in Appendix B.2.4. However, in many cases, we choose to work with a deterministic variant of the problem, or impose a different uncertainty model (as in Chapter 4).

B.3.2.1. UNCONTROLLABLE LOAD

Uncontrollable devices do not offer any flexibility, which makes the behaviour independent from the pricing scheme. This class includes all devices that consume electricity and that do not fit in any of the other categories. We describe the behaviour of these devices with a demand profile $\mathbf{x}_{\square, \mathcal{X}}$. The uncontrollable demand in a household is generally lumped together. Note that if we do not have any household internal cost or restrictions on the capacity of the household electricity connection, then we may also lump together the uncontrollable demand of a complete neighbourhood. Although PV also fits in this category, it is often considered separately. We denote the set of all uncontrollable devices as $\mathcal{I}_{\square}$.

B.3.2.2. TIME SHIFTABLE

Washing machines, dryers and dishwashers can be flexible in the sense that their start time can be shifted within a certain window. We call this class of devices *time shiftables*, and use $\mathcal{I}_{\leftrightarrow}$ to refer to this set of devices. Flex Street time shiftables are similar to EFI time shiftables (Section 5.4.2), but impose fewer constraints. These devices each have to execute a sequence of jobs $J_{\leftrightarrow, \mathcal{N}} = \{1, \dots, n_{J_{\leftrightarrow, \mathcal{N}}}\}$, whereby every job $j \in J_{\leftrightarrow, \mathcal{N}}$ has an arrival time $t_{\leftrightarrow, \mathcal{N}, j, a}$ and a deadline $t_{\leftrightarrow, \mathcal{N}, j, d}$. These jobs correspond to tasks that have to be executed in order by the device. A job can not start before arrival, and must complete before the deadline.¹ We assume that both the arrival time and deadline become certain at the time of arrival. Also, as the jobs have to be executed successively by the device, we may assume that both the arrival times and deadlines of jobs are increasing with the index j (but the job intervals may overlap).

A job consists of one or more segments, and may be interrupted between segments. For example, the dishwasher in Flex Street may be interrupted after 60 and 90 minutes, which leads to three segments (the first 60 minutes, the next 30 minutes, and the final 30 minutes). We let $S_{\leftrightarrow, \mathcal{N}, j} = \{1, \dots, n_{S_{\leftrightarrow, \mathcal{N}, j}}\}$ be the list of segments for job $j \in J_{\leftrightarrow, \mathcal{N}}$. Jobs should contain at least one segment:

$$n_{S_{\leftrightarrow, \mathcal{N}, j}} \geq 1 \quad \forall \mathcal{N} \in \mathcal{I}_{\leftrightarrow}, j \in J_{\leftrightarrow, \mathcal{N}}. \quad (\text{B.2})$$

A segment gives an uncontrollable demand profile of a fixed length. We denote the demand profile of segment s in job j of device $\mathcal{N}$ as $\mathbf{x}_{\leftrightarrow, \mathcal{N}, j, s}$. Every segment s of job j of device $\mathcal{N}$ has a begin time $t_{\leftrightarrow, \mathcal{N}, j, s, b}$, and a completion time $t_{\leftrightarrow, \mathcal{N}, j, s, c}$ (i.e., the segment no longer runs after the completion time):

$$t_{\leftrightarrow, \mathcal{N}, j, s, c} = t_{\leftrightarrow, \mathcal{N}, j, s, b} + |\mathbf{x}_{\leftrightarrow, \mathcal{N}, j, s}| \quad \forall \mathcal{N} \in \mathcal{I}_{\leftrightarrow}, j \in J_{\leftrightarrow, \mathcal{N}}. \quad (\text{B.3})$$

Jobs and segments have to be executed in-order, and may not overlap:

$$t_{\leftrightarrow, \mathcal{N}, j, s, b} \geq t_{\leftrightarrow, \mathcal{N}, j, s-1, c} \quad \forall \mathcal{N} \in \mathcal{I}_{\leftrightarrow}, j \in J_{\leftrightarrow, \mathcal{N}}, s \in \{2, \dots, n_{S_{\leftrightarrow, \mathcal{N}, j}}\}. \quad (\text{B.4})$$

Every job must begin no earlier than the arrival time, and must complete before the deadline:

$$t_{\leftrightarrow, \mathcal{N}, j, 1, b} \geq t_{\leftrightarrow, \mathcal{N}, j, a} \quad \forall \mathcal{N} \in \mathcal{I}_{\leftrightarrow}, j \in J_{\leftrightarrow, \mathcal{N}} \quad (\text{B.5})$$

$$t_{\leftrightarrow, \mathcal{N}, j, (n_{S_{\leftrightarrow, \mathcal{N}, j}}), c} \leq t_{\leftrightarrow, \mathcal{N}, j, d} \quad \forall \mathcal{N} \in \mathcal{I}_{\leftrightarrow}, j \in J_{\leftrightarrow, \mathcal{N}}. \quad (\text{B.6})$$

In a *mixed integer (linear) programming* (MIP) formulation, we refer to these time intervals $t_{\leftrightarrow, \mathcal{N}, j, s, b}$ indirectly through indicators $\mathbf{y}_{\leftrightarrow, \mathcal{N}, j, s}$ for $t \in \mathcal{T}$, which indicate whether $t = t_{\leftrightarrow, \mathcal{N}, j, s, b}$. Note that we have to instantiate these variables only for small range of $\mathcal{T}$, as the start times before the earliest possible start time and after the latest possible start time are infeasible. To determine the bounds of this range, the lengths of the leading and following segments (and, in atypical cases, jobs) are relevant as well.

1. We treat the endtimes in the Flex Street input data as *inclusive* values, and add 1 to obtain the deadline value.

With an assignment of $t_{\leftrightarrow, \mathcal{N}, j, s, b}$ (or $y_{\leftrightarrow, \mathcal{N}, j, s}$), we can now determine the electricity demand vector $\mathbf{x}_{\leftrightarrow, \mathcal{N}}$:

$$x_{\leftrightarrow, \mathcal{N}}(t) = \sum_{j \in J_{\leftrightarrow, \mathcal{N}}} \sum_{s \in S_{\leftrightarrow, \mathcal{N}, j}} \begin{cases} x_{\leftrightarrow, \mathcal{N}, j, s}(t - t_{\leftrightarrow, \mathcal{N}, j, s, b} + 1) & \text{if } t_{\leftrightarrow, \mathcal{N}, j, s, b} \leq t < t_{\leftrightarrow, \mathcal{N}, j, s, c} \\ 0 & \text{otherwise} \end{cases} \quad \forall t \in \mathcal{T}. \quad (\text{B.7})$$

This concludes the definition of a time shiftable device $\mathcal{N}$.

B.3.2.3. EVS/PHEVS

Where time shiftable devices can only be shifted in time, Flex Street assumes that (PH)EVs may be charged at any level $x_{\uparrow, \mathcal{N}}(t)$ between 0 W and a charging limit. We denote the set of (PH)EVs as $\mathcal{I}_{\uparrow}$. Just as time shiftables, these devices have to execute a list of jobs, which we denote as $J_{\uparrow, \mathcal{N}} = \{1, \dots, n_{J_{\uparrow, \mathcal{N}}}\}$. These job $j \in J_{\uparrow, \mathcal{N}}$ also have an arrival time $t_{\uparrow, \mathcal{N}, j, a}$ and a deadline $t_{\uparrow, \mathcal{N}, j, d}$. The jobs of an EV do not overlap. As we can choose from continuous charging levels, it is not necessary to split up the charging in segments. The EVs must always be fully charged between the arrival time and the deadline by a given amount $E_{\uparrow, \mathcal{N}, j}$:

$$\sum_{t \in \{t_{\uparrow, \mathcal{N}, j, a}, \dots, t_{\uparrow, \mathcal{N}, j, d} - 1\}} \tau x_{\uparrow, \mathcal{N}}(t) = E_{\uparrow, \mathcal{N}, j} \quad \forall j \in J_{\uparrow, \mathcal{N}}. \quad (\text{B.8})$$

The case assumes that the charge amount for a given EV is fixed, and PHEVs must always be fully charged as well, i.e. the case does not exploit the hybrid aspect of PHEVs. Furthermore, an EV can only be charged during a job (when the car is parked), and the charge power is limited to charging limit $x_{\uparrow, \mathcal{N}, j}^{\max}$ during a job j :

$$0 \leq x_{\uparrow, \mathcal{N}}(t) \leq \begin{cases} x_{\uparrow, \mathcal{N}, j}^{\max} & \text{if } \exists j : t_{\uparrow, \mathcal{N}, j, a} \leq t < t_{\uparrow, \mathcal{N}, j, d} \\ 0 & \text{otherwise} \end{cases} \quad \forall t \in \mathcal{T}. \quad (\text{B.9})$$

In the Flex Street case, the limit $x_{\uparrow, \mathcal{N}, j}^{\max}$ is equal for all j and $\mathcal{N}$. Also, the case does not consider V2G (i.e. that we may feed in electricity back to the grid), but we may easily include this in the model by allowing for negative values of $\mathbf{x}_{\uparrow, \mathcal{N}}$ in (B.9) during a job.

B.3.2.4. BATTERY STORAGE

Batteries are the most flexible devices in our case, and may charge and discharge at continuous levels. However, a battery has a limited storage capacity, limited charge/discharge power and losses.

Let $\mathcal{I}_{\text{bat}}$ be the set of battery devices. A battery $\mathcal{N} \in \mathcal{I}_{\text{bat}}$ has a SoC variable $s_{\text{bat}, \mathcal{N}}(t)$ for every $t \in \mathcal{T} \cup \{0\}$, which gives the fill level of the battery *after* time interval t . The SoC must always lie between 0 and the maximum SoC $s_{\text{bat}, \mathcal{N}}^{\max}$:

$$0 \leq s_{\text{bat}, \mathcal{N}}(t) \leq s_{\text{bat}, \mathcal{N}}^{\max} \quad \forall t \in \mathcal{T} \cup \{0\} \quad (\text{B.10})$$

The average electricity exchange $x_{\text{bat},\mathcal{N}}(t)$ in time interval t of the battery must lie between the maximum discharge power $x_{\text{bat},\mathcal{N}}^{\min}$ (which is here expressed as a negative power value) and the maximum charging power $x_{\text{bat},\mathcal{N}}^{\max}$:

$$x_{\text{bat},\mathcal{N}}^{\min} \leq x_{\text{bat},\mathcal{N}}(t) \leq x_{\text{bat},\mathcal{N}}^{\max} \quad \forall t \in \mathcal{T} \quad (\text{B.11})$$

Positive values for $x_{\text{bat},\mathcal{N}}(t)$ mean charging (the SoC increases), and negative values mean discharging (the SoC decreases). The SoC now evolves as follows, based on a charge efficiency $\eta_{\text{bat},\mathcal{N}}$:

$$s_{\text{bat},\mathcal{N}}(t) = s_{\text{bat},\mathcal{N}}(t-1) + \tau x_{\text{bat},\mathcal{N}}(t) \cdot \begin{cases} \eta_{\text{bat},\mathcal{N}} & \text{if } x_{\text{bat},\mathcal{N}}(t) \geq 0 \\ 1 & \text{if } x_{\text{bat},\mathcal{N}}(t) < 0 \end{cases} \quad \forall t \in \mathcal{T}. \quad (\text{B.12})$$

Note that the charge efficiency is accounted for at the input, as it would otherwise reduce the effective capacity of the battery. We do not account for the efficiency loss associated with a possible charging and discharging within the same time interval. In some DP formulations, we reduce the maximum discharge level to $x_{\text{bat},\mathcal{N}}^{\min} = -\eta_{\text{bat},\mathcal{N}} x_{\text{bat},\mathcal{N}}^{\max}$ to make the charge and discharge SoC changes symmetric, which reduces the size of the search space. The battery model does not consider losses over time, although these may be added analog to how the losses over time are incorporated in the heating model (Appendix B.3.2.5).

B.3.2.5. HEATING

Households in Flex Street have either a heat pump with an auxiliary resistance heating rod, or gas fired heating (which is not modelled). The heat pumps in Flex Street use a highly simplified thermal model. In the model of Section 3.2, we put the complete heating system of a household in a single node, as there generally is a tight coupling between the operation of the different components. The *Buffer* model of EFI (Section 5.4.3) works similarly and puts a heating system into a single control space, but has a modular model for these type of devices. We define a specific heating system node for Flex Street that contains a heat pump, an auxiliary resistance heating rod, a heat buffer, and the heat demand. We earlier illustrated this node in the lower part of Figure B.1b (p. 276).

Let $\mathcal{I}_h$ be the set of these heating systems, and let $\mathcal{N} \in \mathcal{I}_h$ give an instance of such a system. The heat pump in $\mathcal{N}$ has an electricity demand (input) of $\mathbf{x}_{h,\text{hp},\mathcal{N}}$ over $t \in \mathcal{T}$, and the auxiliary resistance heating rod has an electricity demand of $\mathbf{x}_{h,\text{aux},\mathcal{N}}$. The input power of the heat pump is limited to $x_{h,\text{hp},\mathcal{N}}^{\max}$:

$$0 \leq x_{h,\text{hp},\mathcal{N}}(t) \leq x_{h,\text{hp},\mathcal{N}}^{\max} \quad \forall t \in \mathcal{T}. \quad (\text{B.13})$$

The electricity demand $x_{h,\text{aux},\mathcal{N}}(t)$ of the auxiliary resistance heating rod at time $t \in \mathcal{T}$ is unlimited (yet non-negative):

$$0 \leq x_{h,\text{aux},\mathcal{N}}(t) \quad \forall t \in \mathcal{T}. \quad (\text{B.14})$$

The heat demand consists of heating heat demand (i.e. space heating) and domestic heat demand (hot tap water), which are lumped together in a heat demand vector $\mathbf{d}_{h,\mathcal{X}}$ over $t \in \mathcal{T}$,² and is specified as average thermal power values (W_{th}). Normally (and at least within the considered dataset), the heat demand is positive (in practice, solar radiation may lead to a significant negative heat demand).

The SoC of the heat buffer evolves as follows. Similar to the battery model in Appendix B.3.2.4, the SoC of the buffer $s_{h,\mathcal{X}}$ over $t \in \mathcal{T} \cup \{0\}$ is bounded by 0 and $s_{h,\mathcal{X}}^{\max}$, the maximum SoC:

$$0 \leq s_{h,\mathcal{X}}(t) \leq s_{h,\mathcal{X}}^{\max} \quad \forall t \in \mathcal{T} \cup \{0\}. \quad (\text{B.15})$$

As in Appendix B.3.2.4, $s_{h,\mathcal{X}}(t)$ gives the SoC *after* time interval t . The model considers only energy streams, and not the physical nature of the system (temperatures, flows, etc.). To account for the losses over time, the SoC declines exponentially between time intervals with a factor $\eta_{h,\mathcal{X},\tau} \in (0,1]$, which depends on the time interval length τ . Note that we can determine an equivalent decay factor for a different time interval length τ' by considering n time intervals of length τ and $\frac{\tau}{\tau'}n$ intervals of length τ' :

$$(\eta_{h,\mathcal{X},\tau})^n = (\eta_{h,\mathcal{X},\tau'})^{\frac{\tau}{\tau'}n} \quad (\text{B.16})$$

$$(\eta_{h,\mathcal{X},\tau})^n = (\eta_{h,\mathcal{X},\tau'}^{\frac{\tau}{\tau'}})^n \quad (\text{B.17})$$

$$\eta_{h,\mathcal{X},\tau} = \eta_{h,\mathcal{X},\tau'}^{\frac{\tau}{\tau'}} \quad (\text{B.18})$$

$$(\eta_{h,\mathcal{X},\tau})^{\frac{\tau'}{\tau}} = (\eta_{h,\mathcal{X},\tau'}^{\frac{\tau}{\tau'}})^{\frac{\tau'}{\tau}} \quad (\text{B.19})$$

$$(\eta_{h,\mathcal{X},\tau})^{\frac{\tau'}{\tau}} = \eta_{h,\mathcal{X},\tau'}. \quad (\text{B.20})$$

Note that on several occasions we assume that $\eta_{h,\mathcal{X},\tau} = 1$, i.e. there are no losses over time. The heat pump electricity input contributes to the SoC with a COP factor of $\eta_{h,hp,\mathcal{X}}$, the auxiliary resistance heating rod contributes with a “COP” of $\eta_{h,aux,\mathcal{X}} = 1$, and the demand contributes with a factor -1 , all multiplied by τ to find the accumulated effect over time:

$$s_{h,\mathcal{X}}(t) = \eta_{h,\mathcal{X},\tau} s_{h,\mathcal{X}}(t-1) + \tau(\eta_{h,hp,\mathcal{X}} x_{h,hp,\mathcal{X}}(t) + \eta_{h,aux,\mathcal{X}} x_{h,aux,\mathcal{X}}(t) - d_{h,\mathcal{X}}(t)) \quad \forall t \in \mathcal{T} \quad (\text{B.21})$$

Note that if τ is constant, then we may leave out this constant if we divide $s_{h,\mathcal{X}}^{\max}$ (given in J) by τ (i.e. the buffer capacity is now given in $W\tau$). In principle, the initial SoC of the buffers in Flex Street is undefined, as the length of a year simulation makes these differences negligible. As a default assumption, we take $s_{h,\mathcal{X}}(0) = \frac{1}{2}s_{h,\mathcal{X}}^{\max}$. We use a more evenly spread initial SoC in Section 4.4.

2. In hindsight, this may lead to a significant modelling inaccuracy as heat pumps tend to operate differently for heating and domestic heat demand, due to the large difference in output temperature, i.e. houses with a heat pump often also use low temperature heating for efficiency.

B.3.2.5.1. Integer variant In the case of Appendix B.5 and in Chapter 4, we consider a variant of this problem that restricts the heat pump to be either fully turned on or fully turned off, as many practical heat pumps (e.g. including refrigerators) have this limitation. To model this, we replace (B.13) with:

$$x_{h, hp, \mathcal{N}}(t) \in \{0, x_{h, hp, \mathcal{N}}^{\max}\} \quad \forall t \in \mathcal{T}. \quad (\text{B.22})$$

We do not restrict the operation of the heating rod.

Instead of using an integer variant, we may often use a *pulse width modulation* (PWM) principle to emulate a continuous control (e.g. cycle between turning the device on for 10 minutes and then leaving it off for 20 minutes to obtain an average demand of $\frac{1}{3}x_{h, hp, \mathcal{N}}^{\max}$). However, we should not switch too frequently to avoid damage to the device, i.e. the time intervals should be long enough to use this technique. Also, if we are willing to reintroduce an integer problem, we may assign costs to switch operations. In an implementation, we may minimize the number of switch operations for a given assignment of $x_{h, hp, \mathcal{N}}(t)$ by toggling the state between full-on and full-off once in the middle of a time interval, whereas a naive implementation would turn the device on at the begin of the time interval and then turn it off when sufficient energy has been consumed, which may take two times as many switch operations. For example, if the device is turned off at the end of the first time interval, then we should wait with turning it on exactly until we should leave it on for the rest of the second interval, and then turn it off in the third interval when it has consumed enough. This also helps to distribute the switching operations within a time interval, and thereby leads to less abrupt real-time demand changes. Also, the naive version places most of the demand at the begin of the time interval, whereas the second places the demand at either the begin or end of the interval (still, it may in some cases be worthwhile to spend some extra switches to reduce the synchronization of demand).

B.3.2.5.2. Discrete step model In some DP formulations, it may be worthwhile to introduce a fixed SoC step size (e.g. $\frac{1}{8}x_{h, hp, \mathcal{N}}^{\max}\tau$) to reduce the size of the state space, similar to how we address the battery in Appendix B.3.2.4. We may now express the upper and lower bound of the buffer as a discrete number of SoC steps, which are both shifted upward by the demand over time (Figure 4.10 on p. 168 presents a continuous variant of this concept). More formally, for a SoC step size $s_{h, \mathcal{N}}$ and number of SoC steps $r_{h, \mathcal{N}, s}$ over $\mathcal{T}$ that we have contributed since $t = 0$, we have:

$$r_{h, \mathcal{N}}(t) \in \mathbb{Z} \quad \forall t \in \mathcal{T} \quad (\text{B.23})$$

$$\tau(\eta_{h, hp, \mathcal{N}}x_{h, hp, \mathcal{N}}(t) + \eta_{h, aux, \mathcal{N}}x_{h, aux, \mathcal{N}}(t)) = r_{h, \mathcal{N}}(t) \cdot s_{h, \mathcal{N}} \quad \forall t \in \mathcal{T}, \quad (\text{B.24})$$

while we still have (B.13) and (B.14). Next, we translate the initial SoC and the demand over time to the “step” domain:

$$r_{h,\mathcal{K},\text{lb}}(t) = \left\lceil \frac{-s_{h,\mathcal{K}}(0) + \sum_{t' \in \{1, \dots, t\}} \tau d_{h,\mathcal{K}}(t')}{s_{h,\mathcal{K}}} \right\rceil \quad \forall t \in \mathcal{T} \quad (\text{B.25})$$

$$r_{h,\mathcal{K},\text{ub}}(t) = \left\lceil \frac{s_{h,\mathcal{K}}^{\max} - s_{h,\mathcal{K}}(0) + \sum_{t' \in \{1, \dots, t-1\}} \tau d_{h,\mathcal{K}}(t')}{s_{h,\mathcal{K}}} \right\rceil \quad \forall t \in \mathcal{T} \quad (\text{B.26})$$

$$r_{h,\mathcal{K},\text{s},\text{lb}}(t) \leq r_{h,\mathcal{K},\text{s}}(t) \leq r_{h,\mathcal{K},\text{s},\text{ub}}(t) \quad \forall t \in \mathcal{T}. \quad (\text{B.27})$$

B.4. RESULTS

In this section, we present the results of the simulations with the two control systems of the Flex Street case. To demonstrate the potential of the Flex Street case, the study compared the Intelligator (see Section 2.3.2.1.5) and TRIANA control systems. TRIANA uses IDDP (see Section 3.3) and a minimal real-time control as in Section 2.4, with a planning horizon of around 14 h and a 6 h replanning interval (these parameters are chosen as a tradeoff between the quality of the planning and the execution time of the year simulation). Both control approaches use an existing simulation environment (in our case the simulator developed in [19, 231]), which have both been extended with the Flex Street device set. As the evaluation was performed at the end of 2011, the figures reflect the performance of the DSM control approaches at that time. Note that both approaches have improved significantly since then.

Claessen [48: Fig. 4] presents the demand profiles of the different simulations in each of the considered scenarios (the Pessimistic, Moderate and Optimistic scenario), broken down by device type. The results show that the electricity demand of controllable devices is distributed differently over the day by the two control systems. Intelligator moves most peak demand to the early night, whereas TRIANA spreads demand more evenly over the night. TRIANA responds to the heat demand predictions by filling up the heat buffers in advance of the heat demand peak: in the Moderate scenario, TRIANA can fill up the heat storage and batteries during the drop in demand that is (forecasted to be) caused by PV feed-in around noon. Although the latter leads to a more flat demand profile, betting on PV feed-in may be a more risky strategy, as this feed-in may not materialize, which subsequently leads to a demand peak in the early evening to make up for the demand that has not been served during the afternoon.

The load duration curves at a neighbourhood level are presented in Figure B.3 for the Moderate scenario, considering the two cases with coordination (TRIANA and Intelligator) and the baseline case without coordination. For reference, we have also included the result for the method that we developed in Section 3.5 (although this comparison is not completely fair, as the latter uses more accurate device DPs and considers a deterministic context). A load duration curve is a convenient way to show the amount of time that an electricity system consumes more than a certain demand level (a relevant level may e.g. be a transformer rating). For example, the “peak demand” line marks the required capacity to cover the microgrid’s load for 99.7% of the time (i.e. 1 day per year not covered). On the left hand side of the curve, we see that both control systems give a large decrease in peak demand: the

FIGURE B.3: Load duration curves of TRIANA and Intelligator DSM control approaches. Note that the results represent relatively old (2011) versions of these approaches. We also illustrate the result for a method from this thesis (profile steering) in a deterministic version of the problem.

system	Annual transport losses	
	MV	LV
uncoordinated	€25.84/hh/yr	€21.13/hh/yr
Intelligator	€18.28/hh/yr	€15.37/hh/yr
TRIANA	€17.29/hh/yr	€18.15/hh/yr

TABLE B.3: Line loss transport costs (€0.25/kWh) in simplified LV and MV model in the Moderate scenario.

RPR is 60% for Intelligator and 63% for TRIANA. Furthermore, the middle of the load duration curve is relatively flat for TRIANA, indicating that TRIANA is also good at levelling the demand at less critical off-peak times.

In addition to the peak shaving objective that was given to the control systems as an objective, we now also look at different aspects of the produced demand profiles, following Appendix B.2.5. The objective of peak-shaving the aggregate demand of the microgrid reduces the peak capacity requirements of the MV network. Peak shaving also helps in reducing MV transport losses, due to the quadratic nature of these losses. Table B.3 shows the annual transport costs due to line losses in both the MV and LV networks, based on the simple loss model of Appendix B.2.5. In the results, it can be seen that TRIANA performed better in reducing MV transport losses. Interestingly, Intelligator reduced LV transport losses the best, in particular for households with a large amount of controllability. The results suggest that a tradeoff exists between the optimization of MV losses at an aggregate level, and the optimization of LV losses at a local level within the grid. TRIANA has increased the LV transport losses for houses with battery installations. This results from suboptimal battery planning when

scenario	system	RPR	SPF	DBU	transport cost savings (€/hh/yr)	
					LV	MV
Pessimistic	uncoordinated	(0%)	—	—	(0)	(0)
	Intelligator	32.1%	—	—	0.97	0.94
	TRIANA	32.3%	—	—	0.94	1.15
Moderate	uncoordinated	(0%)	4.25	—	(0)	(0)
	Intelligator	60.2%	4.83	23%	5.76	7.56
	TRIANA	63.3%	4.99	105%	2.98	8.65
Optimistic	uncoordinated	(0%)	4.28	—	(0)	(0)
	Intelligator	66.5%	4.83	44%	12.57	22.54
	TRIANA	70.6%	4.98	107%	2.58	22.51

TABLE B.4: Performance of different control approaches. RPR stands for relative peak reduction, SPF for seasonal performance factor, and DBU for daily battery use.

supply exceeds demand in a house, causing oscillating battery behaviour around PV production hours. A particular shortcoming that results from the randomized prices of IDDP (see Section 3.3) is that it may first let too many batteries charge in a time interval, and then let different batteries discharge to compensate for this overcharging.

In Table B.4, we present an overview of the performance indicators from Appendix B.2.5 that result from the simulations, in each of the scenarios (Pessimistic, Moderate, Optimistic) and for each of the control methods (uncoordinated, Intelligator and TRIANA). TRIANA achieved a slightly higher peak reduction (1 – 6%) than Intelligator in the considered scenarios. The SPF's reveal that TRIANA made the most efficient use of available heat pumps, such that we make limited use of the auxiliary resistance heating rod. The control systems show a significant difference in DBU, but only a marginal difference in savings and abated emissions due to a reduction of transport losses.

Finally, the results of the Flex Street analysis can be used to convince decision makers of the implications of smart grid control in terms of grid infrastructure investments. Based on the aggregate peak demand values over the considered year, Claessen [HAT:8: p. 268] reports the following results, which are illustrated in [HAT:8: Fig. 7]. Without a control system the capacity of the network needs to be increased with respect to the Reference scenario, resulting in extra investment costs for transformers, cables and ditches. Smart grid control decreases the investments by 64% in the Optimistic scenario and even leads to investment savings in the Pessimistic scenario, when a 4% reduction in required investment is obtained with respect to the Reference scenario. Note that the investments are nonlinearly decreasing in the RPR in Table B.4, according to the model of [HAT:8: Equation 8]. The investment savings per household per year are €3.55 in the Pessimistic scenario, €12.57 in the Moderate scenario, and €18.84 in the Optimistic scenario. The savings

mostly come from avoided cables and ditches. The total annual savings resulting from avoided network investments and transport losses in the distribution network range from €5 – 54 per household per year. This figure accounts only for grid related savings and not yet for other possible sources of profit, e.g. from participation in the market.

B.4.1. DISCUSSION

The presented analysis method is designed to compare simulations of control systems, aiming to provide an effective way to compare DSM control approaches from different research groups. However, the approach is for now limited to simulations. A comparison of real implemented control systems would be able to address other operational aspects, such as the running time of computations and the bit rate of communication. Even so, the current method can be improved by strengthening the Flex Street model in at least two ways.

First, Flex Street uses bottom-up modelling of residential demand profiles. The effort of initially modelling many components is rewarded by creating a flexible tool for the analysis of smart grid control systems. Based on known or assumed characteristics of residents and devices, it easily allows constructing different profiles for houses. The ability to adjust such specific design parameters facilitates the design of new standardized cases.

In contrast to the above, top-down datasets generally convey very little information on their composition, which hinders design revisions. Generating new cases is more convenient when a bottom-up approach is used in the entire model. Flex Street currently uses just two top-down datasets: one for uncontrollable electricity demand and one for PV feed-in.

Second, additional constraints can easily be added to Flex Street. Since e.g. transport losses in the LV network appear to be in the same order of magnitude as those in the MV network, these should be defined more clearly in the model to allow control systems to take these losses into account. Moreover, the specific network topology and local characteristics can be included in the model to obtain meaningful results for specific cases. Constraints limiting the power on internal lines, or LV transport costs, can then also be incorporated in the objective function of a control system. Next to line losses, peak demand on LV lines should be considered as an important parameter, determining investment costs more accurately.

Finally, a number of factors have not been incorporated in the model, that would be necessary for a full analysis of costs and benefits. On the one hand, these include investment costs of the required ICT infrastructure and operating costs due to e.g. the increased wear of batteries. On the other hand, additional benefits may result from trading electricity or ancillary services on the electricity market, which may further increase the incentive for smart grid control. For an economic evaluation of the results, both of these considerations should be taken into account.

B.4.2. CONCLUSION

A comparison of the Intelligator and TRIANA control systems demonstrates that our analysis method is able to quantitatively compare the performance of these control systems. Using the presented Flex Street model, three case variants were designed with various penetration rates of controllable and renewable devices within a neighbourhood microgrid. In each case, the control system is given the objective to perform peak shaving on the aggregate electricity demand of the microgrid over a one year period. The simulation results show that performance differences between the control methods are relatively small for this objective.

The Intelligator has been able to generate a relative peak reduction of 32 – 67% over the year, depending on the amount of controllability offered by the device population in each scenario. TRIANA has achieved a peak reduction of 32 – 71%. Both control systems lead to comparable savings in transport losses in the MV network. Depending on the case at hand, the control resulted in cost savings of €1 – 23 per household per year and CO₂ savings of 0.25 – 6 kg per household per year.

In the simulations discussed here, the control systems had not been optimized to take into account transport losses in the LV network. However, our analysis shows that such losses may be significant and, therefore, should be taken into account. This could be implemented either by adjusting model constraints, or as part of a control system's objective function. The scenarios with various penetration rates of controllable and renewable devices have shown that peak shaving can lead to benefits through reduced transport losses and network investments in the distribution network, totalling €5 – 54 per household per year.

Mutual learning is a benefit of comparative analysis, which is demonstrated by ongoing research at VITO and the University of Twente. Current research on the Intelligator system is oriented towards incorporating planning in its business agent function, which should further increase performance. A self-learning business agent, able to adapt to seasonal effects using reinforcement learning, has already been developed. The developers of TRIANA are researching strong and weak points of auction based real-time control, comparing it to TRIANA's current real-time control strategy based on integer linear programming. The integration of new optimization techniques and network constraint handling is already in progress.

The Flex Street model provides a flexible tool for comparative analysis of smart grid control systems. Our analysis method is able to compare simulations of independently developed control systems using standardised cases with a broad scope. Flex Street facilitates case studies resembling a realistic setting for the operation of smart grid control systems. However, the method does not address quantitative indicators which may be relevant for a system implementation, such as run times of algorithms and communication speeds.

Flex Street can be improved by adding constraints on power lines or device usage to increase the complexity of the control task. However, simplicity remains a good property. Finally, it is possible to cover the entire model through bottom-up modelling of consumption/production, which would further increase the flexibility of Flex Street as a tool for smart grid researchers.

B.5. A CASE STUDY BASED ON FLEX STREET

B.5.1. INTRODUCTION

The Flex Street gives a bottom-up scenario description that can easily be tailored to reflect a different scenario. Lazopoulou [205] developed a variant of Flex Street to study and quantify the possible profit that DSM may achieve in a pilot project [97]. This case study considers a group of 500 households, but in this case only few of these households have controllable devices. The case considers a centralized PV park in an urban context, with a connected central EV charging facility.

For this case, the focus shifts from balancing to optimization for the market. Several pricing schemes are considered: two of these pricing schemes consider individual households (*time of use* (TOU) and *real time pricing* (RTP)), and two schemes consider the group as a whole (*critical peak pricing* (CPP) and *critical peak pricing with rebate* (CPR)). The latter cases also integrate accounting for a capacity tariff for the whole neighbourhood, which accounts for the maximum of the demand in a month and in a year. Note that these pricing schemes have been introduced conceptually in Section 2.3.1.4. We have performed simulations with TRIANA on this case to determine the expected demand profiles for the scenarios, using models of the pricing schemes.

B.5.2. SCENARIO

The neighbourhood model is illustrated in Figure B.4. We have a neighbourhood where n_{ctl} out of 500 households have some controllable devices. Next to this, all 500 households contribute to the uncontrollable loads, and we have a PV park with a co-located EV charging point with n_{EV} shared EVs. The EVs are the dominant source of flexibility in the neighbourhood. In principle, the EVs can be charged for free with the PV feed-in, and a 50% energy tax discount (which makes up about two-thirds of the Dutch residential electricity price) is obtained if we use the PV production within the neighbourhood.

The case study considers three scenarios with different penetration rates of controllable devices, PV penetration, and electricity prices, which are called 2013, 2020-I, and 2020-II. The scenario 2013 represents the initial scale of a PV park pilot project; 2020-I and 2020-II represent a 10- and 20-fold increase in scale of the 2013 scenario. The parameters for the scenarios are given in Table B.5.

The case study uses a device set with parameters that differ from the described Flex Street case. In addition, the case we consider here introduces refrigerators and freezers. They are both implemented within the heat pump model with binary control (on and off) and a static demand pattern.

To optimize this case, we use the column generation approach (Section 3.4), as it has the flexibility to incorporate various objectives. To reduce the computational effort of a year simulation, we use time intervals of a large length ($\tau = 3600$ s), with a 48 h planning horizon and a 12 h replanning interval.

In the model presented in Figure B.4, we put the PV park and the neighbourhood into a single column generation node. To model the possible profit from PV self-consumption, we use a simplified piecewise linear function for the cost on the

FIGURE B.4: Model of case study.

scenario	PV peak	PV/year	UD/hh/yr	n_{ctl}	n_{WM}	n_{TD}	n_{DW}	n_{RF}	n_{FR}	n_{EV}
2013	28.1 kWp	22733 kWh	2750 kWh	10	9	6	5	10	5	2
2020-I	280.8 kWp	227330 kWh	2617 kWh	100	99	72	60	99	50	20
2020-II	516.6 kWp	454660 kWh	2617 kWh	200	198	144	120	198	100	40

TABLE B.5: Parameters of scenarios in Appendix B.5 [205: p. 28]. The field UD/hh/yr gives the uncontrollable demand per household per year. The right half of the table refers to the number of controllable washing machines (n_{WM}), tumble dryers (n_{TD}), dishwashers (n_{DW}), refrigerators (n_{RF}), freezers (n_{FR}), and EVs (n_{EV}).

neighbourhood electricity connection, where all demand below a certain threshold (normally chosen at a level of 0 W) has a price of €/kWh, which motivates the self-consumption of PV: if we do not consume there, then we have to consume somewhere with positive prices, which is more expensive.³

An interesting aspect of this case is the capacity tariff for the large customer variants of the case. For this tariff, we assume that we have to pay €1.90/kW for the monthly maximum demand in a time interval (in principle 15 min), and €14.00/kW for the contracted maximum over the year [307: Tabel 4]. We assume that the contracted maximum corresponds with the actual maximum value. To put these values in perspective, if we consider the 2013 uncontrollable demand in Table B.5 with an electricity price of €0.07/kWh and a peak-to-average ratio of 2 in every month, then the electricity and transport itself costs €96250, and the capacity costs $313.9 \text{ kW} \cdot (12 \cdot 1.90 + 14.00) = €11552$. Even though the commodity costs are still dominant, there is a substantial incentive for peak reduction at times of high demand. As discussed in Section 3.5.2, this means that we should account for the maximum demand outside of the planning horizon at hand. For this, we carry the

3. At the time when the experiments were performed, it was hard to integrate new objectives without compromising the structure of the software implementation. The current implementation can describe the problem more directly, i.e. with a nested structure and the earlier described cost function.

FIGURE B.5: Maximum-so-far in case study (2013, CPP).

maximum observed demand between planning sessions (with a safety margin of 10 kW), and reset the value at the end of the month. Note that we should account for cases where the first part of the planning session is in one month, and the other part in the following month, as we may then inadvertently carry the value from the current month to the next month. We may resolve this case by resetting the value whenever the planning horizon contains a month boundary.

B.5.3. RESULTS

The outcome of the simulations are reported in [205: Chapter 6]. The DSM control gives a peak reduction of 0.8% (2013), 18% (2020-I), and 30% (2020-II) in the large customer cases. The distance between 2020-I and 2020-II indicates that some saturation effects are already visible (i.e. two times as much flexibility leads to less than two times as much peak reduction), although it is still meaningful to add more flexibility. The cost reduction ranges between €0.43 and €37.57 per household per year in different scenarios, including the households that do not own controllable devices (in the last case, 40% of the households have controllable devices). The work notes that this means that the budget for DSM infrastructure is limited if we still want to make a profit.

One of the observed problems with the EVs in the considered scenario is that the EVs are actually being used during the day. An EV can not be charged by the PV park while it is being used, meaning that the case mostly uses the neighbourhood PV tax deduction instead of charging the EVs. Note that this problem applies in general for shared EVs with a high utilization during the day. Instead, the EVs are charged during periods with low prices (i.e. the night and in the early morning in the considered price dataset). The profits may increase substantially in the high PV penetration scenarios if we are able to charge the EVs during the day, or have more storage options to make effective use of the PV feed-in.

In Figure B.5, we report the maximum observed value of the month and of the year for the 2013 case with the CPP tariff. In this case, demand is significantly higher in winter than in summer, which may be attributed in part to the PV production in summer. A (positive) side effect of this is that the annual maximum value is already (almost) settled at the beginning of the year, which means that we no longer have unnecessary cost to keep this value low. As the demand goes down on average in the first half of the year, we see a similar trend with the monthly maximum demand. However, in the second half of the year, we see that the approach continually has to adapt the maximum, which means that we may have constrained the problem too much at the beginning of the month. This may lead e.g. to less exploitation of periods with low prices, as we want to avoid an increase of the monthly peak value.

B.6. AGGREGATE LOWER BOUND MODEL

The Flex Street case leads to a very large scenario. However, for the evaluation of the control methods it may be useful if also the optimal solutions to the problem are known. We may try to model the complete scenario directly, e.g. as a (relaxed) MIP. However, as the scenario has 1560 devices and $4 \cdot 24 \cdot 365$ time intervals and we may expect to use on average at least 2 variables to model the behaviour of a device in a time interval, we obtain a problem with 109324800 variables, which is intractable or at least very challenging without some modelling improvements. Therefore, it is difficult to determine the optimal control solution for such a large scenario.

We propose a method to determine a lower bound for (some variants of) the problem. To this end, we make an abstraction of the available flexibility, based on the constraint aggregation principle of Section 2.3.2.1.4 (Figure 2.1, [331]). Our contribution is a mapping of Flex Street to this model, and the interpretation of this model as a lower bound for the problem at hand. We lump each device class into a single representative device that describes the aggregate energy flexibility of these devices in this class. The representative device has at least as much flexibility as the original devices have together, and can therefore be used to provide a lower bound for (variants of) the original optimization problem. Note that although we may in principle lump all devices (of all classes) into a single representative device, this may lead to decisions that are not executable in practice, e.g. we may create situations where a heat pump is supposed to discharge electricity from the heat buffer, which is physically impossible. Note that we can no longer distinguish between the consumption of households, and that we again obtain a large problem if we account for the households separately.

B.6.1. AGGREGATE MODEL

The representative device $\mathcal{x}$ has the following structure. Let $\mathbf{P}_{\mathcal{x}}^{\text{ub}}$ give the upper bound and $\mathbf{P}_{\mathcal{x}}^{\text{lb}}$ the lower bound on the electricity demand of $\mathcal{x}$ over $\mathcal{T}$. Next, $\mathbf{E}_{\mathcal{x}}^{\text{ub}}$ gives the upper bound and $\mathbf{E}_{\mathcal{x}}^{\text{lb}}$ the lower bound on the aggregate electricity demand

over $\mathcal{T}$ (after this time interval). The energy content is measured as if we store electricity. Now, the demand $\mathbf{x}_\mathcal{X}$ over time has the constraints:

$$P_\mathcal{X}^{\text{lb}}(t) \leq x_\mathcal{X}(t) \leq P_\mathcal{X}^{\text{ub}}(t) \quad \forall t \in \mathcal{T}. \quad (\text{B.28})$$

For the energy constraints, we introduce an internal SoC variable $s_\mathcal{X}(t)$ for $t \in \mathcal{T} \cup \{0\}$, with $s_\mathcal{X}(0) = 0$:

$$s_\mathcal{X}(t) = s_\mathcal{X}(t-1) + \tau x_\mathcal{X}(t) \quad \forall t \in \mathcal{T} \quad (\text{B.29})$$

$$E_\mathcal{X}^{\text{lb}}(t) \leq s_\mathcal{X}(t) \leq E_\mathcal{X}^{\text{ub}}(t) \quad \forall t \in \mathcal{T}. \quad (\text{B.30})$$

We can aggregate a set of devices $\mathcal{I}$ into such a device $\mathcal{X}$ by adding up the bounds:

$$P_\mathcal{X}^{\text{ub}}(t) = \sum_{i \in \mathcal{I}} P_i^{\text{ub}}(t) \quad \forall t \in \mathcal{T} \quad (\text{B.31})$$

$$P_\mathcal{X}^{\text{lb}}(t) = \sum_{i \in \mathcal{I}} P_i^{\text{lb}}(t) \quad \forall t \in \mathcal{T} \quad (\text{B.32})$$

$$E_\mathcal{X}^{\text{ub}}(t) = \sum_{i \in \mathcal{I}} E_i^{\text{ub}}(t) \quad \forall t \in \mathcal{T} \quad (\text{B.33})$$

$$E_\mathcal{X}^{\text{lb}}(t) = \sum_{i \in \mathcal{I}} E_i^{\text{lb}}(t) \quad \forall t \in \mathcal{T}. \quad (\text{B.34})$$

The aggregate problem is less constrained than the sum of individual problems: for example, a battery may have a low storage capacity and a high discharge rate, and an EV has a high storage capacity but no discharge (V2G) ability. In this case, the aggregate model has the option to discharge the lumped energy that is provided by the EV, which is not feasible in the original problem. This means that we can not always determine a solution for $\mathcal{I}$ from a solution to $\mathcal{X}$, but use this solution only as a lower bound.

B.6.2. DEVICE MODELS

To use the aggregate model of Appendix B.6.1, we have to describe each of the devices in terms of the aggregate model. Note that we will reuse $\mathcal{X}$ to refer to these devices, and follow the notation from Appendix B.3. In the following, we address each of the normal device types in Flex Street.

B.6.2.1. UNCONTROLLABLE LOAD

An uncontrollable load $\mathcal{X} \in \mathcal{I}_\square$ has no storage capacity and a demand of $\mathbf{x}_\mathcal{X}$:

$$\mathbf{P}_\mathcal{X}^{\text{lb}} = \mathbf{P}_\mathcal{X}^{\text{ub}} = \mathbf{x}_\mathcal{X}. \quad (\text{B.35})$$

Although the uncontrollable load in principle has no energy storage, to keep the model consistent we have to specify:

$$E_\mathcal{X}^{\text{lb}}(t) = E_\mathcal{X}^{\text{ub}}(t) = \sum_{t' \in \{1, \dots, t\}} \tau x_\mathcal{X}(t') \quad \forall t \in \mathcal{T}. \quad (\text{B.36})$$

B.6.2.2. TIME SHIFTABLE

A time shiftable $\mathcal{N} \in \mathcal{I}_{\leftrightarrow}$ has a maximum demand that is determined by the active job and a minimum demand of 0 (unless the job's segments have negative demand values). At most one job and segment can be active at a time. The minimum and maximum aggregate energy consumption are determined by the energy content of the jobs that may have started and must have been completed, respectively. First, we introduce an auxiliary definition $J_{\leftrightarrow, \mathcal{N}, t}$ that gives the potentially active jobs at time t :

$$J_{\leftrightarrow, \mathcal{N}, t} = \{j \in J_{\leftrightarrow, \mathcal{N}} : t_{\leftrightarrow, \mathcal{N}, j, a} \leq t < t_{\leftrightarrow, \mathcal{N}, j, d}\} \quad \forall t \in \mathcal{T}. \quad (\text{B.37})$$

The flexibility space is now given as follows:

$$p_{\mathcal{N}}^{\text{lb}}(t) = \begin{cases} \min(0, \min_{j \in J_{\leftrightarrow, \mathcal{N}, t}, s \in S_{\leftrightarrow, \mathcal{N}, j}} \min \mathbf{x}_{\leftrightarrow, \mathcal{N}, j, s}) & \text{if } J_{\leftrightarrow, \mathcal{N}, t} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad \forall t \in \mathcal{T} \quad (\text{B.38})$$

$$p_{\mathcal{N}}^{\text{ub}}(t) = \begin{cases} \max_{j \in J_{\leftrightarrow, \mathcal{N}, t}, s \in S_{\leftrightarrow, \mathcal{N}, j}} \max \mathbf{x}_{\leftrightarrow, \mathcal{N}, j, s} & \text{if } J_{\leftrightarrow, \mathcal{N}, t} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad \forall t \in \mathcal{T}. \quad (\text{B.39})$$

The energy upper bound is determined by all jobs that may ever have been active:

$$E_{\mathcal{N}}^{\text{ub}}(t) = \sum_{j \in \bigcup_{t' \in \{1, \dots, t\}} J_{\leftrightarrow, \mathcal{N}, t'}} \sum_{s \in S_{\leftrightarrow, \mathcal{N}, j}} \sum_{\mathbf{x} \in \mathbf{X}_{\leftrightarrow, \mathcal{N}, j, s}} \tau \mathbf{x} \quad \forall t \in \mathcal{T}. \quad (\text{B.40})$$

The energy lower bound is almost equal to the upper bound, except that it excludes the jobs that may be active at time t :

$$E_{\mathcal{N}}^{\text{lb}}(t) = \sum_{j \in (\bigcup_{t' \in \{1, \dots, t-1\}} J_{\leftrightarrow, \mathcal{N}, t'}) \setminus J_{\leftrightarrow, \mathcal{N}, t}} \sum_{s \in S_{\leftrightarrow, \mathcal{N}, j}} \sum_{\mathbf{x} \in \mathbf{X}_{\leftrightarrow, \mathcal{N}, j, s}} \tau \mathbf{x} \quad \forall t \in \mathcal{T}. \quad (\text{B.41})$$

B.6.2.3. EVS/PHEVS

For (PH)EVs $\mathcal{N} \in \mathcal{I}_{\uparrow}$, we obtain a definition that is almost the same as Appendix B.6.2.2, except that the jobs already have the desired energy flexibility structure:

$$J_{\uparrow, \mathcal{N}, t} = \{j \in J_{\uparrow, \mathcal{N}} : t_{\uparrow, \mathcal{N}, j, a} \leq t < t_{\uparrow, \mathcal{N}, j, d}\} \quad \forall t \in \mathcal{T} \quad (\text{B.42})$$

$$p_{\mathcal{N}}^{\text{lb}}(t) = 0 \quad \forall t \in \mathcal{T} \quad (\text{B.43})$$

$$p_{\mathcal{N}}^{\text{ub}}(t) = \begin{cases} x_{\uparrow, \mathcal{N}, j}^{\max} & \text{if } J_{\uparrow, \mathcal{N}, t} = \{j\} \text{ for some } j \\ 0 & \text{otherwise} \end{cases} \quad \forall t \in \mathcal{T} \quad (\text{B.44})$$

$$E_{\mathcal{N}}^{\text{ub}}(t) = \sum_{j \in \bigcup_{t' \in \{1, \dots, t\}} J_{\uparrow, \mathcal{N}, t'}} E_{\uparrow, \mathcal{N}, j} \quad \forall t \in \mathcal{T} \quad (\text{B.45})$$

$$E_{\mathcal{N}}^{\text{lb}}(t) = \sum_{j \in (\bigcup_{t' \in \{1, \dots, t-1\}} J_{\uparrow, \mathcal{N}, t'}) \setminus J_{\uparrow, \mathcal{N}, t}} E_{\uparrow, \mathcal{N}, j} \quad \forall t \in \mathcal{T}. \quad (\text{B.46})$$

B.6.2.4. BATTERY STORAGE

A battery $\mathcal{N} \in \mathcal{I}_{\text{bat}}$ has an energy content of $s_{\text{bat},\mathcal{N}}^{\max}$ (offset by the initial SoC $s_{\text{bat},\mathcal{N}}(0)$) and a power range of $x_{\text{bat},\mathcal{N}}^{\min}$ to $x_{\text{bat},\mathcal{N}}^{\max}$:

$$P_{\mathcal{N}}^{\text{ub}}(t) = x_{\text{bat},\mathcal{N}}^{\min} \quad \forall t \in \mathcal{T} \quad (\text{B.47})$$

$$P_{\mathcal{N}}^{\text{ub}}(t) = x_{\text{bat},\mathcal{N}}^{\max} \quad \forall t \in \mathcal{T} \quad (\text{B.48})$$

$$E_{\mathcal{N}}^{\text{lb}}(t) = -s_{\text{bat},\mathcal{N}}(0) \quad \forall t \in \mathcal{T} \quad (\text{B.49})$$

$$E_{\mathcal{N}}^{\text{ub}}(t) = s_{\text{bat},\mathcal{N}}^{\max} - s_{\text{bat},\mathcal{N}}(0) \quad \forall t \in \mathcal{T} \quad (\text{B.50})$$

We can not model the charge efficiency $\eta_{\text{bat},\mathcal{N}}$ with the aggregate model, as the losses would be seen as actual charging (the electricity consumption from the losses also leads to an increase of the SoC). We circumvent this with a similar separate aggregate battery model that is aware of the efficiency factor, provided that the efficiency is the same for all batteries or there are only a few types.

B.6.2.5. HEATING

The heating model of Appendix B.3.2.5 is comparable with the earlier devices, except that the required progress in accumulated energy demand is now determined by the heat demand and by the COP of the heat pump.

We are not able to account for losses, nor for the use of the auxiliary resistance heating rod, because these both affect the effective COP and thereby change the relation between electricity demand and the accumulated energy demand over time (however, the COP may vary over time). These aspects may be accounted for with a specialized model (as discussed for the battery storage in Appendix B.6.2.4), or the effective COP may be estimated iteratively. Although the use of the auxiliary resistance heating rod is necessary in some cases to make the model feasible at an individual household level, this should not be visible in the aggregate model. Therefore, we may choose not to model the auxiliary resistance heating rod to find a lower bound for the case at hand.

This gives the following heating model for a heating system $\mathcal{N} \in \mathcal{I}_{\text{h}}$, where we do not model the auxiliary resistance heating rod:

$$P_{\mathcal{N}}^{\text{lb}}(t) = 0 \quad \forall t \in \mathcal{T} \quad (\text{B.51})$$

$$P_{\mathcal{N}}^{\text{ub}}(t) = x_{\text{h,hp},\mathcal{N}}^{\max} \quad \forall t \in \mathcal{T} \quad (\text{B.52})$$

$$E_{\mathcal{N}}^{\text{lb}}(t) = \left(-s_{\text{h},\mathcal{N}}(0) + \sum_{t' \in \{1, \dots, t-1\}} \tau d_{\text{h},\mathcal{N}}(t') \right) / \eta_{\text{h,hp},\mathcal{N}} \quad \forall t \in \mathcal{T} \quad (\text{B.53})$$

$$E_{\mathcal{N}}^{\text{ub}}(t) = \left(-s_{\text{h},\mathcal{N}}(0) + s_{\text{h},\mathcal{N}}^{\max} + \sum_{t' \in \{1, \dots, t\}} \tau d_{\text{h},\mathcal{N}}(t') \right) / \eta_{\text{h,hp},\mathcal{N}} \quad \forall t \in \mathcal{T}. \quad (\text{B.54})$$

Note that if we model microCHPs, the “COP” in terms of electricity demand is negative, and the electricity storage develops in the negative direction.

B.6.3. RESULTS

B.6.3.1. ENERGY FLEXIBILITY SPACES

Before we consider any optimization, we look at the energy flexibility spaces of the devices for the Flex Street Moderate scenario. We consider the flexibility spaces for the first week (in winter) and the 27th week (in summer) of the year. Note that the first day of the first week may have some start-up effects. For the summer week, we reset the accumulated energy values to make it comparable with the winter week.

- B.6.3.1.1. Uncontrollable load** Figure B.6 illustrates the flexibility space of the uncontrollable load (including PV). The upper plots give the flexibility in electricity demand power value, and the lower plots give the flexibility in accumulated electricity demand over time. These loads do not have any flexibility. The plots show that the uncontrollable load in summer is generally much lower than in winter. Also, it shows that the PV penetration is modest, as the sum of uncontrollable load and the PV feed-in rarely becomes negative. The PV feed-in is by far not enough to supply all load, especially in winter where we have a deficit of about 30 MWh per week (≈ 10 kWh per household per day) if we only consider the uncontrollable load.
- B.6.3.1.2. Time shiftable** Figure B.7 gives the flexibility space of the time shiftable devices, i.e. all washer/dryers and dishwashers are aggregated to a single resource. In the lower plots we see that these devices offer about 12 to 20 h of shifting flexibility, which is given by the horizontal distance between the lines. In the upper plots we see that the charge rate of the aggregate device has an upper limit of 170 to 280 kW, depending on the time of day. Looking at the vertical distance between the lines in the lower plot, this maximum can be sustained for about $\frac{600 \text{ kWh}}{200 \text{ kW}} = 3 \text{ h}$, which corresponds with the washer/dryer program length. Comparing winter (a) and summer (b), there are no meaningful differences between the winter and summer for the time shiftable devices in the scenario.
- B.6.3.1.3. EVs/PHEVs** In Figure B.8, the flexibility space of the PHEVs is given. These devices together have about $2^{1/2}$ times as much energy content as the time shiftables, and a maximum demand of 740 kW. However, the PHEVs do not offer flexibility to shift the demand outside of the night period. Therefore, the PHEVs are the dominant source of controllable demand in the evening and in the night. Again, the winter and summer flexibility are virtually the same in the scenario.
- B.6.3.1.4. Battery storage** Figure B.9 shows the flexibility space of the aggregate battery. As only 20 households have a battery, the total provided power and energy are rather low. However, (according to the model) batteries have an infinite temporal shifting flexibility, and are the only devices that offer negative demand. Based on this, we can already determine a lower bound on the peak demand of the scenario: we know that the batteries have a total minimum demand of -68 kW, and the uncontrolled load has a maximum demand value of 371.6 kW, giving a lower bound of 303.6 kW if the energy content (width) of the peak and the other controllable

FIGURE B.6: Aggregate energy flexibility of uncontrollable load in Flex Street.

FIGURE B.7: Aggregate energy flexibility of time shiftables in Flex Street.

FIGURE B.8: Aggregate energy flexibility of PHEVs in Flex Street.

FIGURE B.9: Aggregate energy flexibility of batteries in Flex Street.

FIGURE B.10: Aggregate energy flexibility of heating in Flex Street.

FIGURE B.11: Comparison of peak minimization and quadratic solutions.

loads are not relevant. Note that the results achieved with control for this case lead to a peak demand of at least 467.1 kW (see Figure 3.22), which is 53.9% higher than this (very optimistic) lower bound value.

B.6.3.1.5. Heating Figure B.10 illustrates the flexibility of the aggregate heating system. In winter (a), we see that the heating system gives a significant contribution to the demand, yet is very limited in its temporal shifting flexibility, in particular in the evening where the uncontrollable load is high as well. The buffer is dimensioned quite small in comparison to the amount of load in winter. At the sixth day (between 5 and 6 in (a)) we see a large heat demand peak that leads to a sustained high (and thereby barely shiftable) aggregate electricity demand with an average of $\frac{2000 \text{ kWh}}{16 \text{ h}} = 125 \text{ kW}$, which may already explain most of the earlier mentioned gap between the lower bound and the actual control result. In summer (b), the heating system becomes a much less relevant load. However, due to the low heat demand, the temporal shifting flexibility becomes high ($\approx 1\frac{1}{2}$ day).

B.6.3.2. PEAK DEMAND

To determine a lower bound on the minimum peak demand, we optimize:

$$\min \max \mathbf{x}_{\mathcal{N}}, \quad (\text{B.55})$$

where $\mathcal{N}$ refers to a node that sums the demand of the above described aggregate energy flexibility models (one model per device class). Solving this year problem with CPLEX takes 6.55 s on a desktop PC with an Intel Core i5-6600 (4 cores, 16 GB of RAM) if we choose to use the simplex method, and 2.52 s with the barrier (interior point) method. We find a lower bound on the minimum peak demand of 442.1 kW. Note that even though both optimization methods find the same peak demand value, the structure of the solutions differs substantially, as we show in Appendix B.6.3.4. This value is in line with what we expected from the shape of the energy flexibility spaces in Appendix B.6.3.1 and the achieved values in Figure 3.22. The peak demand is 5.7% higher with the best performing method in Figure 3.22 than the bound indicates. We believe that at least half of this difference may be attributed to the inaccuracy of the heating abstraction, as it does not account for the forced use of the auxiliary resistance heating rod at some (often critical) times. Indeed, if we account for this by postponing the heating rod use as long as possible, then we obtain a peak value of 463.0 kW, which decreases the gap to 0.9%. As the quadratic solution (using (B.57) from Appendix B.6.3.4) has a flat demand profile and does not charge the battery within the relevant critical period (which would introduce extra losses), this bound should still be accurate. In the following we use this revised model to get some further insight.

B.6.3.3. RAMP RATE

To determine a lower bound on the ramp rate (here measured in W/15 min as we do not have more detailed information), we optimize:

$$\min \max_{t \in \{2, \dots, n_t\}} |x_{\mathcal{N}}(t) - x_{\mathcal{N}}(t-1)|. \quad (\text{B.56})$$

Solving this problem takes 8.17 s, and we find a maximum value of 6.2 kW/15 min, which is much lower than what we find in Figure 3.23. We have two explanations for this. First, the considered case has multiple objectives. If we for example add (B.55) to our objective with a relative weight of 10 for (B.55) and a weight of 1 for (B.56) (i.e. peak minimization is far more important), then the maximum ramp rate remains virtually the same (6.3 kW), although the data shows that the average demand over the year is 42 W higher due to battery losses. If we add costs for these losses in the objective, then the peak demand remains the same, but the maximum ramp rate increases to 12.8 kW/15 min.

B.6.3.4. QUADRATIC OBJECTIVE

Whereas peak minimization decreases the highest demand value, it does not motivate a general flattening of demand. To achieve this, we consider a quadratic objective:

$$\min \tau \mathbf{x}_N^\top \mathbf{x}_N. \quad (\text{B.57})$$

Solving this problem takes 3.90 s. The proposed solution has a maximum demand of 463.0 kW, which is equal to the solution for the peak minimization objective in Appendix B.6.3.2. Although the peak demand is equal, the demand profiles resulting from (B.55) and from (B.57) are substantially different. Figure B.11a gives the demand profile on the days near to the unavoidable peak demand value. The figure shows that the formulation of the optimization problem may have a significant impact on the structure of the solution. The specific solution method used may have an impact as well: the peak minimization with the simplex method leads to a very unsmooth demand profile, as it focuses only on achieving a low peak but besides that does not put some effort in achieving a “nice” profile with respect to some other (non-specified) criteria. This unsmooth behaviour motivates the introduction of various auxiliary objectives to address such undesired behaviour (i.e. the peak minimization objective does not represent that this unsmoothness is undesired). In contrast, the peak minimization with the barrier method leads to a more smooth demand profile, although the peak demand value is still reached more often than necessary (again, the objective does not represent that this is undesirable). Objective (B.57) leads to a demand profile that clearly avoids the demand peak value, and gives a highly flattened demand profile. However, it is also characterized by rapid demand changes (which may be prevented by auxiliary objective terms). This type of multiple plateau behaviour is also observed in [191: Fig. 2] (which considers a quadratic optimization of a battery) and in [MT:2: Figure 4] (which considers the linear optimization of a battery with a secondary objective of minimizing the sum of absolute differences between demand values). The corresponding load duration curves in Figure B.11b demonstrate that these trends persist over the year.

B.6.3.5. QUADRATIC RAMP RATE

As an alternative, we may also consider an objective where we try to minimize the ramp rate quadratically:

$$\min \sum_{t \in \{2, \dots, n_t\}} (x_N(t) - x_N(t-1))^2. \quad (\text{B.58})$$

This leads to a very smooth demand profile over time (marked diff. QP in Figure B.11). The maximum ramp rate increases to 10.8 kW/15 min in comparison to 122.0 kW/15 min for the case in Appendix B.6.3.3. It is interesting that this profile also performs quite well from a peak shaving perspective (+ 2.2%), even though the objective does not make this goal explicit.

B.7. CONCLUSION

Flex Street is a flexible bottom-up framework for the comparison of DSM control methods and the generation of DSM scenarios. We find that it is very convenient as a base for the evaluation of control methods. We internally use this framework extensively, and have several implementations for different simulation tools. The specification of device models, rather than imposing a specific implementation, allows the reuse of the devices in different environments, which has allowed the use of existing simulators of VITO and the University of Twente (i.e. the simulator developed in [19, 231]). Based on this, simulation results were obtained that enabled a straightforward comparison of the control systems. Also, the model could be adapted for use in an economic study. Although Flex Street is meant to be relatively simple to make it portable across coordination methods, one of the consequences of simplicity is that it ignores many physical aspects. This may make the outcomes of the simulation less accurate for some applications.

In this thesis, we extend the work of [BTT:8] and [205] by contributing a mathematical model for the devices in the Flex Street case. To help in the evaluation of control methods, we developed an aggregate model of the simulation case that can be used as a lower bound model for the case. This model is applied for the evaluation of the control methods from Chapter 3. This model proves to be convenient and efficient, although it by design does not describe local phenomena that may be relevant in some cases (e.g. household level demand and local costs).

B.7.1. RECOMMENDATIONS

B.7.1.1. BETTER UNCERTAINTY MODELLING

The uncertainty modelling of Flex Street, as described in Appendix B.2.4, uses a fairly simplistic profile-based model of uncertainty. A better approach may be to describe the model that is used to generate the realizations of the profile, and leave the forecasting of the realization to the control approach.

B.7.1.2. MORE PRECISE OBJECTIVE

At the moment, the objective has been defined in terms of separate KPIs, whereas it would be better to have a clear objective function that we can optimize for. This ambiguity should be resolved in a future revision of Flex Street.

IMPLEMENTATION DETAILS

coordination in time 307 · space 312 · practice 316

• • • • •

C.1. COORDINATION IN TIME

C.1.1. EXPLOITING MULTIPLE COMMODITY PROVIDERS (QUADRATIC CASE)

In Section 3.2.4.5 (p. 61), we discuss methods to optimize the consumption from multiple commodity providers. Here, we consider the case where each of the commodity providers has a quadratic upward cost function. For this, consider a minimization problem:

$$\min a_\alpha x_\alpha^2 + b_\alpha x_\alpha + a_\beta x_\beta^2 + b_\beta x_\beta + z^{\text{sub}}, \quad (\text{C.1})$$

with z^{sub} the internal cost of the problem subtree, and $x = x_\alpha + x_\beta$ (the other internal constraints within the problem are not given). As $x_\alpha = x - x_\beta$, we get:

$$\min a_\alpha (x - x_\beta)^2 + b_\alpha (x - x_\beta) + a_\beta x_\beta^2 + b_\beta x_\beta + z^{\text{sub}} \quad (\text{C.2})$$

$$= \min a_\alpha x^2 - 2a_\alpha x x_\beta + a_\alpha x_\beta^2 + b_\alpha x - b_\alpha x_\beta + a_\beta x_\beta^2 + b_\beta x_\beta + z^{\text{sub}} \quad (\text{C.3})$$

$$= \min a_\alpha x^2 + b_\alpha x + (a_\alpha + a_\beta) x_\beta^2 + (b_\beta - b_\alpha - 2a_\alpha x) x_\beta + z^{\text{sub}}. \quad (\text{C.4})$$

Now, for a given x , we may minimize (C.4) as a function of x_β . For this, we first prune everything that does not depend on x_β , leading to:

$$\min (a_\alpha + a_\beta) x_\beta^2 + (b_\beta - b_\alpha - 2a_\alpha x) x_\beta, \quad (\text{C.5})$$

which has its minimum at:

$$x_\beta = \frac{-b_\beta + b_\alpha + 2a_\alpha x}{2(a_\alpha + a_\beta)}. \quad (\text{C.6})$$

Substituting (C.6) back in (C.4) gives:

$$\min a_\alpha x^2 + b_\alpha x + (a_\alpha + a_\beta) \left(\frac{-b_\beta + b_\alpha + 2a_\alpha x}{2(a_\alpha + a_\beta)} \right)^2 + (b_\beta - b_\alpha - 2a_\alpha x) \frac{-b_\beta + b_\alpha + 2a_\alpha x}{2(a_\alpha + a_\beta)} + z^{\text{sub}} \quad (\text{C.7})$$

$$= \min a_\alpha x^2 + b_\alpha x + \frac{1}{2} \frac{((-b_\beta + b_\alpha) + 2a_\alpha x)^2}{2(a_\alpha + a_\beta)} + ((b_\beta - b_\alpha) - 2a_\alpha x) \frac{(-b_\beta + b_\alpha) + 2a_\alpha x}{2(a_\alpha + a_\beta)} + z^{\text{sub}}. \quad (\text{C.8})$$

We simplify this by substituting $\gamma = -b_\beta + b_\alpha$. This gives:

$$= \min a_\alpha x^2 + b_\alpha x + \frac{1}{2} \frac{(\gamma + 2a_\alpha x)^2}{2(a_\alpha + a_\beta)} + (-\gamma - 2a_\alpha x) \frac{\gamma + 2a_\alpha x}{2(a_\alpha + a_\beta)} + z^{\text{sub}} \quad (\text{C.9})$$

$$= \min a_\alpha x^2 + b_\alpha x + \frac{1}{2} \frac{\gamma^2 + 4a_\alpha x \gamma + 4a_\alpha^2 x^2}{2(a_\alpha + a_\beta)} - \frac{\gamma^2 + 4a_\alpha x \gamma + 4a_\alpha^2 x^2}{2(a_\alpha + a_\beta)} + z^{\text{sub}} \quad (\text{C.10})$$

$$= \min a_\alpha x^2 + b_\alpha x - \frac{1}{2} \frac{\gamma^2 + 4a_\alpha x \gamma + 4a_\alpha^2 x^2}{2(a_\alpha + a_\beta)} + z^{\text{sub}}. \quad (\text{C.11})$$

$$= \min a_\alpha x^2 + b_\alpha x - \frac{\frac{1}{4}\gamma^2 + a_\alpha x \gamma + a_\alpha^2 x^2}{a_\alpha + a_\beta} + z^{\text{sub}} \quad (\text{C.12})$$

$$= \min \left(a_\alpha - \frac{a_\alpha^2}{a_\alpha + a_\beta} \right) x^2 + \left(b_\alpha - \frac{a_\alpha \gamma}{a_\alpha + a_\beta} \right) x - \frac{\frac{1}{4}\gamma^2}{a_\alpha + a_\beta} + z^{\text{sub}}, \quad (\text{C.13})$$

from which we can drop the constant term $-\frac{\frac{1}{4}\gamma^2}{a_\alpha + a_\beta}$. Using $\gamma = -b_\beta + b_\alpha$, we get:

$$= \min \left(a_\alpha - \frac{a_\alpha^2}{a_\alpha + a_\beta} \right) x^2 + \left(b_\alpha - \frac{a_\alpha b_\alpha - a_\alpha b_\beta}{a_\alpha + a_\beta} \right) x + z^{\text{sub}}, \quad (\text{C.14})$$

which gives our combined cost function. On a first view, the coefficient of x^2 looks as if it may become non-positive in some cases, which would change the structure of the problem. However, for $a_\alpha, a_\beta > 0$, we get $0 < \frac{a_\alpha^2}{a_\alpha + a_\beta} < a_\alpha$, implying that the coefficient $(a_\alpha - \frac{a_\alpha^2}{a_\alpha + a_\beta})$ is positive.

To retrieve the assignments for the original problem from a solution x , we may first use (C.6) to find x_β , and then $x_\alpha = x - x_\beta$. If we do have restrictions on x_α and x_β , then often a piecewise model may be used: at some point, we know that we should consume the minimum or maximum amount of that provider. We note that the problem of distributing demand over multiple commodity suppliers is strikingly similar to the problem of distributing demand over multiple time intervals, which

means that the techniques to address these problems may be used interchangeably. For example, the algorithm of [190], which was originally developed to efficiently spread load over time intervals, may be used to efficiently spread the load over different commodities, for a given amount of total demand.

C.1.2. LOCAL SEARCH FOR PATTERN COMBINATIONS¹

C.1.2.1. INTRODUCTION

In the following, we propose an algorithm by which an aggregator can “cherry-pick” the best combination of patterns from an available pattern set for each children, following the model of Section 3.2. The presented optimization methods combine the solutions of subproblems to determine the solution for an aggregate problem. Finding the optimal combination of patterns is usually hard, as the number of possible combinations grows combinatorially with the number of patterns that is available for each subproblem. Pattern selection by MIP or *mixed integer quadratic programming* (MIQP) is often prohibitively expensive, or impractical due to nonlinearities in the problem. Therefore, we propose a local search strategy that is inexact yet computationally tractable to find good combinations of patterns. The approach resembles a metaheuristic/Monte Carlo optimization method. In contrast to the MIP or MIQP approach, the problem does not need to have a linear or a quadratic structure. The objective function of the problem should be reasonably smooth to ensure that the local search is effective. Constraints are in principle not supported, but may be approximated using the objective function. We need efficient functions to effect a pattern change and to compute the objective evaluation function.

C.1.2.2. ALGORITHM

The algorithm starts from an initial best solution candidate, which may be obtained randomly, from a previous solution, by greedy selection, or some other method. For robustness against local minima, making multiple starts is possible, although we have never observed this to be necessary in practice. Next, the following process iteratively refines the solution. In every iteration, we select n_c out of n subproblems, where n is the total number of subproblems. For each of these n_c subproblems, we select a random pattern from the available solution set, and apply this selection to the main problem. Next, we determine the objective value. If the objective value improves, then the current subproblem solution assignment replaces the best solution candidate, else we go back to the best candidate. The search ends after a cutoff criterion is met, e.g. after a chosen fixed number of iterations n_k , when the allotted time runs out, or when an objective value threshold is met.

For a fixed n_k , the number of operations is deterministic, provided that the bootstrap, update, evaluation and rollback operations are deterministic (which is usually true).

1. This algorithm was proposed in internal communication by Marco Gerards.

FIGURE C.1: Example of an iteration of the local pattern combination search algorithm that finds an improving pattern combination ($n = 4$, $n_v = 10$, $n_c = 2$).

The algorithm follows these steps:

- Bootstrap – choose the initial best solution candidate;
- Evaluate initial solution;
- While cutoff criterion not met:
 - Choose n_c distinct subproblems;
 - For these n_c subproblems:
 - * Choose a random replacing pattern (from the same subproblem, with replacement);
 - * Continue with next iteration if none of the patterns changes (if running time determinism is not needed);
 - * Update – apply swap to solution candidate.
 - Evaluate – determine objective value
 - If objective value improves:
 - * Commit – set current solution as best solution.
 - Else:
 - * Rollback – apply reverse swap to best solution.

C.1.2.3. EXAMPLE

Figure C.1 gives an example of an iteration of the algorithm. In this example, the algorithm successfully proposes an improving pattern switch set (typical progress is not as fast, in particular for later iterations). The variables represent the (abstract)

electricity demand profiles of a group of devices, which add up to the electricity profile of a building (analogously, the demand of a group of building determines the demand of a neighbourhood). The goal is to minimize the squared distance to an all-zero profile (if the demand volume is fixed, then this is a flattening objective).

The algorithm randomly chooses subproblems 3 and 4, and randomly selects the fourth and the second pattern as the swap candidates, respectively. The objective value before the swap is 50. The swaps subtract the values of subproblem 3, pattern 3 and subproblem 4, pattern 1, and add the values of subproblem 3, pattern 4 and subproblem 4, pattern 2 to the master pattern (other, less straightforward mappings are possible). The swaps result in an almost flat profile. The objective value of the proposed pattern is 42, which makes the proposed pattern an improving pattern ($42 < 50$): we keep this assignment as the best-known pattern.

C.1.2.4. EXPLORATION VERSUS EXPLOITATION

The choice for the value of n_c is a tradeoff between exploration and exploitation. A higher value of n_c is more likely to choose an unsuitable solution for a subproblem, but a low value of n_c may fail to discover some combinations due to local minima. We recommend to start at $n_c = 1$, and increase the value progressively. Values up to $n_c = 2$ or 3 work well in practice. Alternatively, instead of taking $n_c > 1$, the algorithm may be changed to continue with a non-improving pattern change with some probability. The theoretical advantage of this alternative is that in the limit, eventually the optimal solution is chosen. The same effect can be reached with $n_c = n$, although this works less well in the average case. However, these alternatives are essentially inefficient ways to enumerate all combinations. The practical disadvantage of the first alternative is that bookkeeping becomes more complex.

The pattern selection algorithm relies on an efficient update operation and an efficient evaluation function, because these are applied in every iteration. In many cases, these steps can be performed independent of the number of subproblems (however, note that the useful number of iterations does depend on the number of subproblems).

C.1.2.5. SWAP OPERATION

The update/swap operation determines the effect of swapping the patterns of n_c subproblems (this operation may be separated in separate steps). For simple balance problems, the procedure works as follows: subtract the contribution of each of the original patterns, and add the contribution of each of the patterns that replace the original patterns. The same operation can be used as the inverse operation, but then with the replacement parameters (or the add/subtract operations) swapped. This operation can be done in $\mathcal{O}(n_c \cdot n_v)$, where n_v is the number of balance variables (n_v is usually a multiple of n_t according to the number of considered commodities; if necessary, computing the effect on the balance variables can be part of a preprocessing step).

A (constant-factor) reduction in the number of operations per iteration is possible by precomputing each of the pattern difference applications, with a cost of $\mathcal{O}(n \cdot n_v \cdot n_p^2)$ in time and in space (with n_p the number of patterns for each subproblem). However, this step is usually not worthwhile, because it increases memory use considerably, and assumes that $n \cdot n_p^2 \ll n_c \cdot n_k$, which does often not hold.

C.1.2.6. PATTERN ROLLBACK

Almost all generated candidate solutions are discarded. To be able to revert the solution, the implementation can write the new solution to a temporary array, apply the inverse pattern change afterwards or compute the new candidate in-place without storing it. If the solution is the new best solution, it should be recomputed. The in-line approach can save memory operations, but relies on a specialized evaluation function that can apply the requested pattern swap.

C.2. COORDINATION IN SPACE

C.2.1. PARAMETERS OF METAHEURISTICS EXPERIMENTS

The evaluation in Section 4.5 of the metaheuristic optimization approach of Chapter 4 uses the following parameter assignment:

» *Supply offer strategy representation* (Section 4.2.3)

We want a small representation so as to reduce the search effort, but we should still be able to express a good assignment.

- Time points $n_{ts} = 9$

A large number of time points allows us to find a strategy that adapts more quickly (e.g. to price changes over time), but also increases the search space and thereby increases the search effort. For a fixed iteration count, a larger number of time points may thereby lead to a less good solution. As a balance, we choose $n_{ts} = 9$ time points, which gives one point for every 3 hours on a 24 h horizon (at 0:00, 3:00, ..., 24:00).

- Priority levels $n_{ps} = 3$

The same holds for the priority levels: a large number of priority points allows to specify a more detailed supply offer curve, but increases the search space. We give the supply offer quantity for a (very) low, normal, and high priority level, and interpolate the values in between.

» *Assignment of supply offer strategy variables* (Section 4.3)

In all experiments, we use metaheuristic optimization to assign the variables of the supply offer strategy. We use the following common parameters for each of the metaheuristic optimization algorithms:

- Iteration count $n_k = 300$

The iteration count should be large enough to let the search algorithm converge. For the chosen values of n_{ts} and n_{ps} , the remaining improvement of $\tilde{z}(\hat{\mathbf{x}}^\pi)$ after $k = 300$ is only marginal. Note that even though $\tilde{z}(\hat{\mathbf{x}}^\pi)$ still slightly decreases, this hardly translates to a corresponding decrease in $z(\hat{\mathbf{x}}^\pi)$ for large k .

- Initial assignment $\hat{\mathbf{x}}^{(0)}$

For the case at hand, we use $\hat{x}_{t'}(p') = \{0, 1/2 \sum_{i \in \mathcal{I}^d} x_i^{\max}, \sum_{i \in \mathcal{I}^d} x_i^{\max}\}$ for all $t' \in \{1, \dots, n_{ts}\}$ and $p' \in \{1, \dots, n_{ps}\}$. The intuition behind this vector is that a device should run at full power when the priority is very high, and should not run when the priority is very low. Without considering the cost function at hand, this seems to give a reasonable neutral result vector from which the search can start. Possible alternatives to find a suitable value for $\hat{\mathbf{x}}^{(0)}$ are to estimate it from history, by (dispatch unaware) optimization, or from pilot simulations ahead of the actual search. In a rolling horizon approach where we revise the planning e.g. every 6 hours, we may reuse the part of $\hat{\mathbf{x}}^{(0)}$ that corresponds to the optimization time period at hand.

- Mutation operation: (4.21)

- Step size $\sigma = |\mathcal{I}^d| \cdot 800 \text{ W}$

The mutation operators need to be able to make changes that are of the same order of magnitude as the maximum electricity demand of the cluster. We found that $\sigma = 8000 \text{ W}$ performs well for $|\mathcal{I}^d| = 10$.

- Change probability $p_+ = 30\%$

Here, we have to make a tradeoff between search locality (which changes very few parameters) and a more global search (which changes more parameters).

Next, we consider the specific algorithms:

- Random optimization (Section 4.3.2)

This algorithm has no further parameters.

- Simulated annealing (Section 4.3.3)

- Acceptance parameter $c_{\text{accept}} = 2$

A lower value gives a higher probability that a suboptimal solution is selected.

- Genetic algorithm (Section 4.3.4)

The mutation operation here differs from the others in that we first determine the “best known” solution from multiple earlier solutions using (4.28), and then apply (4.21) to that “best known” solution. The parameters following from (4.28) are:

- Initial population $\mathcal{X}_{is}^{(0)} = \{\hat{\mathbf{x}}^{(0)}\}$

Where the previous two algorithms start from a single initial assignment $\hat{\mathbf{x}}^{(0)}$, the genetic algorithm may instead be seeded with an initial population set $\mathcal{X}_{is}^{(0)}$. We choose to populate this set only with the initial assignment as given before.

- Maximum population size $n_{\mathcal{X}} = 5$
A higher population size gives more diversity in the solution set from which we derive solutions, but may also hamper progress in finding good new candidates as we hold on to solutions with a poor objective value.
- Mutation operation parent count $n_{\mathcal{X},\text{gen}} = 2$
Using more parents may be beneficial to combine orthogonal traits of the parents, but may also suppress “recessive” traits.

» *Use case* (Section 4.4)

For each of the configurations of the supply offer strategy presented above, we consider our use case with the following parameters:

- Device configuration

We derive our case from the Flex Street case, and consider only the heat pumps. This case offers many parameters for heat pumps. We take default values for all of these parameters, as already presented in Section 4.4:

- Number of heat pumps $|\mathcal{I}^d| = 10$
We choose the number of heat pumps large enough such that we see “aggregate” behaviour, yet also not too large so as to limit simulation effort.
- Buffer size $s_i^{\max} = 8 \text{ kWh}$
A larger buffer size leads to more flexibility and a possible higher robustness to forecast errors, but also increases expectations as set by the reference optimal solution.
- Coefficient of performance $\eta_i = 5$
- Buffer losses over time = 0 %/h
We do not consider heat losses over time, or other SoC-dependent losses. Other losses should be accounted for by the COP or by the heat demand.
- Maximum electricity input of heat pump $x_i^{\max} = 2000 \text{ W}$
On top of this maximum input, we may use an auxiliary resistance heating rod (with a “COP” of 1) to meet heat demand in periods of very high demand, or when the buffer is not sufficiently pre-charged before a demand peak event. In the device bidding functions, the heating rod is made unavailable until the buffer is near empty.
- Heat pump electricity input control range
In principle, we assume that we can switch a heat pump to any electricity consumption level between 0 W and 2000 W. However, many residential heat pumps can only be turned on and off. We consider both of these cases:
 - * Continuous: $x_i(t) \in [0, 1]$
 - * Binary (on/off): $x_i(t) \in \{0, 1\}$
In this case, we also allow partial runs of the heat pump (i.e. for a duration less than τ) until the buffer is completely full.

- Objective/cost function

We choose the cost function (as well as the rest of the problem, i.e. the device model) in such a way that we can determine a reference optimal solution with a *quadratic programming* (QP) formulation of the problem.

- Electricity prices $\mathbf{c}$, $\tilde{\mathbf{c}}$

We use the electricity prices from two days $\mathbf{c}_a$ and $\mathbf{c}_b$ with a distinct price profile over time. In particular, the prices in $\mathbf{c}_b$ are much higher than in $\mathbf{c}_a$, and have a different distribution of high- and low-price periods over the day. We later show $\mathbf{c}_a$ in Figure 4.11 and Figure 4.13, and show $\mathbf{c}_b$ in Figure 4.15. In both cases, we assume that the forecasted prices are equal to the real prices: $\tilde{\mathbf{c}} = \mathbf{c}$.

- Balancing term $\beta = 10^{-11}/|I^d| \in \text{W}^{-2} \text{s}^{-1}$

To discourage excessive demand peaks, we add a quadratic penalty term to the cost function. This term is chosen such that it represents about half of the cost in the reference optimal solution. We divide the balancing term by $|I^d|$ to compensate for the quadratic growth of this cost term.

- “Sell-out” penalty: (4.42)

To maximize profit, the use of an optimization approach with a finite horizon should lead to empty buffers at the end of the considered time horizon to avoid (the cost associated with) as much electricity consumption as possible. This is undesirable in practice, and therefore we penalize both a too high and a too low SoC at the end of the horizon, which should ideally be equal to the SoC at the begin of the horizon. The penalty is based on the highest and lowest price in $\mathbf{c}$ (and during the planning in $\tilde{\mathbf{c}}$). We consider the aggregate SoC of all devices, as it is difficult to control the SoC of individual buffers.

- Device bidding strategy for heat pumps

Every device needs a demand bidding strategy to participate in the auction. These bidding strategies determine which combinations of control actions are available, and may thereby affect the performance. We choose a single bidding strategy for all heat pumps. We consider the following bidding strategies:

- SoC-based bidding strategy, dynamic flexible range (Section 4.4.6.2)

The bidding curve follows directly from the SoC of the associated buffer.

- n/m bidding strategy (Section 4.4.6.4)

The bidding strategy adapts its bids according to (an estimate of) the future average demand.

Due to unsatisfactory results, we have not further pursued the SoC-based bidding strategy with a static flexible range (Section 4.4.6.1) and the l bidding strategy (Section 4.4.6.3), and have therefore not included these cases in the simulation study. We nevertheless mention these strategies to stress that these device level bidding strategies impact what we can achieve with our control.

- Heat demand forecast and possible realizations

The heat demand of the households that are served by the heat pumps is in principle uncertain, and we model this uncertainty with a scenario based approach. Each forecast scenario $\tilde{\xi} \in \tilde{\Xi}$ defines a heat demand vector $\mathbf{d}_i^{\tilde{\xi}}$ for every household corresponding to $i \in \mathcal{I}^d$. After planning based on $\tilde{\Xi}$, we use our supply offer strategy with multiple possible realizations $\xi \in \Xi$ with a heat demand vector $\mathbf{d}_i^{\xi}$, and measure the cost. We take the heat demand from individual days in the Flex Street dataset to make up both the forecast and realization scenarios.

- Deterministic variant

The deterministic variant of the problem assumes that the heat demand forecast corresponds with the realized heat demand.

- * $n_{\Xi} = n_{\tilde{\Xi}} = 1$

In this case, we use the first day in the dataset (Figure B.2a) to determine the heat demand for both the forecast and the realization.

- Stochastic variant

The stochastic variant of the problem assumes that the heat demand forecast does not correspond with the possible realizations of the heat demand. We consider three different cases:

- * $n_{\Xi} = n_{\tilde{\Xi}} = 7$

In this case, we take the first 7 days of the dataset (Figure B.2a...g) as the forecast scenarios $\tilde{\Xi}$, and the following 7 days (Figure B.2h...n) as the possible realizations Ξ . The average heat demand in Ξ is 30% higher than forecasted in $\tilde{\Xi}$.

- * $n_{\Xi} = n_{\tilde{\Xi}} = 10$

We take $\tilde{\Xi}$ as the first 10 days of the dataset (Figure B.2a...j), and Ξ as the following 10 days (Figure B.2k...t). The average heat demand in Ξ is 33.5% higher than forecasted in $\tilde{\Xi}$.

- * $n_{\Xi} = n_{\tilde{\Xi}} = 7$ (swapped Ξ and $\tilde{\Xi}$)

In this case, we use the same days as for $n_{\Xi} = n_{\tilde{\Xi}} = 7$, but swap the roles of $\tilde{\Xi}$ and Ξ : we use the first 7 days for Ξ , and the following 7 days for $\tilde{\Xi}$. This gives a case where we overestimate the demand during the planning.

C.3. COORDINATION IN PRACTICE

C.3.1. TRIANA-EF-PI PROTOCOL FOR DECENTRALIZED CONTROL

To connect the TRIANA aggregators and clients, we developed a simple request/response message protocol. The protocol uses the Protocol Buffers library [139], from which we can automatically generate protocol handlers for most popular programming languages. The protocol is in principle portable across multiple implementations of TRIANA or compatible optimization software. So far, we have only used this capability during development, in a custom command-line Python

tool that uses the generated library to exchange formatted messages with the software. The tool can serve both as a client and a server, and facilitates basic communication (e.g. basic automatic message generation, shortcuts for standard demand patterns, etc.).

We have also considered a plain text protocol, because communication efficiency is currently not considered as a dominant factor; binary packing is not essential at the moment. Nevertheless, Protocol Buffers provides various message parsing and generation services “for free”, and provides bindings for most major programming environments. The protocol design uses simple optimizations where this does not seriously complicate its design or implementation (e.g. not sending buffers full of zeroes); however, it makes no attempt to be as small as possible. This may be reconsidered when switching to more constrained communication services, e.g. *wireless sensor networks* (WSNs), *power line communication* (PLC), GPRS, or LoRaWAN. In such cases, revising the message encoding and the protocol interaction patterns to use the communication medium more efficiently are probably worthwhile.

The protocol describes the following message types. Note that for message request/response pairs, the request messages are postfixed by Q and the responses are postfixed by A:

- HelloQ (client $\rightarrow$ server) and HelloA (server $\rightarrow$ client): Registers a client at a server. The nodes exchange the identity and the capabilities with the node on the other end. The client announces its (static) commodity port mapping. The plan-based coordination of demand requires time synchronization. The nodes check that the clocks of the nodes are (almost) synchronized. Synchronization is important to ensure that the planned time intervals of the different nodes correspond to each other. We assume that connections are relatively short-lived (shorter than a day), such that a synchronization check on connection set-up is sufficient to account for clock skew. Ensuring a limited lifetime is trivial (e.g. by breaking the connection after some randomized time). Nodes may synchronize using a common external *network time protocol* (NTP) source or a distributed clock synchronization algorithm. We acknowledge that the protocol is vulnerable to time synchronization attacks, e.g. [377].
- PlanQ (s $\rightarrow$ c): Proposes a pricing scheme for a given period of time. For synchronization reasons, the time period may have a small overlap with the past. A list of port and type annotated, run-length encoded coefficients describes the cost function. We currently support linear prices and quadratic cost coefficients.
- PlanA (c $\rightarrow$ s): Gives the pattern response for a PlanQ request, matched by the plan ID. A list of port and type annotated, run-length encoded coefficients describes the pattern. The coefficients give the expected values of the power (in W) or volume flow (in m³/s) for each time interval, depending on the type of port.

- `CommitPlanQ` ($s \rightarrow c$) and `CommitPlanA` ($c \rightarrow s$): Requests the client to follow an earlier determined `Plan` pair. When the client agrees, the plan subsumes earlier plans in the same time period. Plans before and after the time period remain unchanged, although the new plan may change the outcome of these plans. The client can implicitly reject the plan by refusing to answer.
- `UpdateQ` ($s \rightarrow c$) and `UpdateA` ($c \rightarrow s$): Asks the client to give a current estimate of the commodity consumption profile over a defined horizon. The server can ask for a single report, or a periodical reporting. The client may also send unsolicited updates, which may motivate the server to start a re-planning.
- `Oops` ($s \leftrightarrow c$): A general error message, with a status code and a message string. To avoid infinite message loops, an `Oops` must never give a message response.

This concludes the protocol description, and also the appendices of this thesis. $\mathfrak{S}$

REFERENCES

- [1] ACM. “Analyse van de Autoriteit Consument en Markt met betrekking tot de voorgenomen afspraak tot sluiting van 80er jaren kolencentrales in het kader van het SER Energieakkoord”. *Dutch*. SEP 2013. <https://www.acm.nl/nl/download/publicatie/?id=12033> (⌚ MAY 29, 2015). p. 241
- [2] ADELMAN Daniel & MERSEREAU AJ. “Relaxations of weakly coupled stochastic dynamic programs”. In *Operations research* 56 : 3, 2008, pp. 712–727. DOI 10.1287/opre.1070.0445. p. 32
- [3] AGENTSCHAP NL. “Couperus smart grid – smart grid pilot projects”. SEP 2013. http://www.rvo.nl/sites/default/files/2013/09/Couperus%20Smart%20Grid%20UK_0.pdf (⌚ MAY 8, 2015). pp. 247, 271
- [4] ALBADI Mohamed H & EL-SAADANY EF. “Demand response in electricity markets: an overview”. In *Power engineering society general meeting (PES GM)*. JUN 2007. DOI 10.1109/PES.2007.385728. p. 262
- [5] ALLIANDER. “EF-Pi: an open source platform”. DEC 2014. <https://www.youtube.com/watch?v=LQQHVnJqjPc> (⌚ MAY 25, 2016). p. 198
- [6] ALTAY Can & DELİÇ H. “Distributed energy management of microgrids with Dantzig–Wolfe decomposition”. In *ISGT Europe*. IEEE PES, OCT 2014. DOI 10.1109/ISGTEurope.2014.7028972. p. 76
- [7] ALUR Rajeev & DILL DL. “A theory of timed automata”. In *Theoretical computer science* 126 : 2, 1994, pp. 183–235. ISSN 0304-3975. DOI 10.1016/0304-3975(94)90010-8. p. 209
- [8] ALVEY Trevor, GOODWIN D, MA X et al. “A security-constrained bid-clearing system for the New Zealand wholesale electricity market”. In *Power systems, IEEE transactions on* 13 : 2, MAY 1998, pp. 340–346. ISSN 0885-8950. DOI 10.1109/59.667349. pp. 10, 13, 15
- [9] ANDERSEN Anders N & SORKNÆS P. “Next generation, CHP to balance intermittent production from renewables”. *Cogeneration & On-Site Power Production magazine*. JUL 2011. <http://www.cospp.com/articles/print/volume-12/issue-4/features/next-generation-chp-to-balance-intermittent-production-from-renewables.html> (⌚ MAR 31, 2015). p. 241
- [10] ANDRESEN Tino. “Europe gas carnage shown by EON closing 3-year-old plant”. *News article*. MAR 2013. <http://www.bloomberg.com/news/articles/2013-03-12/europe-gas-carnage-shown-by-eon-closing-3-year-old-plant-energy> (⌚ MAY 29, 2015). p. 241
- [11] APPELRATH Hans-Jürgen, TERZIDIS O & WEINHARDT C. “Internet of Energy”. In *Business & information systems engineering* 4 : 1, SP Gabler Verlag, 2012. ISSN 2363-7005. DOI 10.1007/s12599-011-0197-x. p. 260
- [12] von APPEN Jan, BRAUN M, STETZ T et al. “Time in the sun: the challenge of high PV penetration in the German electric grid”. In *Power and energy magazine* 11 : 2, IEEE PES, MAR 2013, pp. 55–64. ISSN 1540-7977. DOI 10.1109/MPE.2012.2234407. p. 18

- [13] APX POWER NL. "Day-ahead auction". Web site. <https://www.apxgroup.com/trading-clearing/day-ahead-auction/> (⌚ JUN 8, 2015). pp. 12, 13
- [14] APX POWER UK. "Spot market products". Brochure. JAN 2013. https://www.apxgroup.com/wp-content/uploads/ProductSheets_APX_Power_UK.pdf (⌚ JUN 8, 2015). p. 13
- [15] AUSTIN ENERGY. "Appliance efficiency program – air conditioning". <http://powersaver.austinenergy.com/wps/portal/psp/residential/offerings/cooling-and-heating/air-conditioning> (⌚ APR 6, 2015). p. 247
- [16] AUSUBEL Lawrence M. "An efficient ascending-bid auction for multiple objects". In *American economic review* √94 : 5, DEC 2004, pp. 1452–1475. DOI 10.1257/0002828043052330. p. 26
- [17] AUSUBEL Lawrence M. "An efficient dynamic auction for heterogeneous commodities". In *American economic review* √96 : 3, JUN 2006, pp. 602–629. DOI 10.1257/aer.96.3.602. p. 27
- [18] BABA Norio, SHOMAN T & SAWARAGI Y. "A modified convergence theorem for a random optimization method". In *Information sciences* √13 : 2, 1977, pp. 159–166. ISSN 0020-0255. DOI 10.1016/0020-0255(77)90026-3. p. 155
- [19] BAKKER Vincent. "TRIANA: a control strategy for smart grids". PhD thesis. University of Twente, JAN 2012. ISBN 978-90-365-3314-0. DOI 10.3990/1.9789036533140. pp. 31–34, 46, 49–51, 62, 66, 67, 70, 72, 117, 133, 252, 265, 268, 280, 288, 306
- [20] BAKKER Vincent, NYKAMP S, REINELT J et al. "Controlling and optimizing of energy streams in local buildings in a field test". In *22nd international conference and exhibition on electricity distribution (CIRED)*. Stockholm, Sweden, JUN 2013. ISBN 978-1-84919-732-8. DOI 10.1049/cp.2013.0771. pp. 30, 191, 200
- [21] BALLING Lothar. "Fast cycling and rapid start-up: new generation of plants achieves impressive results". In *Modern power systems* √31 : 1, 2011, pp. 35–41. p. 9
- [22] BANKS John, MASSY K, WOOD L et al. "Assessing the role of distributed power systems in the U.S. power sector". Brookings Institution Energy Security Initiative, OCT 2011. http://www.brookings.edu/~media/research/files/papers/2011/10/distributed-power-systems/10_distributed_power_systems.pdf (⌚ JUN 4, 2015). p. 254
- [23] BARNHART Cynthia, JOHNSON EL, NEMHAUSER GL et al. "Branch-and-price: column generation for solving huge integer programs". In *Operations research* √46 : 3, INFORMS, 1998, pp. 316–329. pp. 78, 81, 86
- [24] BECKER Thomas A, SIDHU I & TENDERICH B. "Electric vehicles in the United States: a new model with forecasts to 2030". Tech. report 2009.1.v2.0. Center for Entrepreneurship & Technology, University of California, Berkeley, AUG 2009. p. 248
- [25] BENDERS Jacques F. "Partitioning procedures for solving mixed-variables programming problems". In *Numerische Mathematik* √4 : 1, Springer, DEC 1962, pp. 238–252. ISSN 0029-599X. DOI 10.1007/BF01386316. p. 22
- [26] BEN-TAL Aharon & NEMIROVSKI A. "Robust solutions of uncertain linear programs". In *Operations research letters* √25 : 1, Elsevier, AUG 1999, pp. 1–13. ISSN 0167-6377. DOI 10.1016/S0167-6377(99)00016-4. p. 16
- [27] BERTSIMAS Dimitris, LITVINOV E, SUN XA et al. "Adaptive robust optimization for the security constrained unit commitment problem". In *Power systems, IEEE transactions on* √28 : 1, FEB 2013, pp. 52–63. ISSN 0885-8950. DOI 10.1109/TPWRS.2012.2205021. p. 16
- [28] BERTSIMAS Dimitris & SIM M. "The price of robustness". In *Operations research* √52 : 1, INFORMS, FEB 2004, pp. 35–53. ISSN 0030-364X. DOI 10.1287/opre.1030.0065. p. 16
- [29] BERTSIMAS Dimitris & TSITSIKLIS JN. "Introduction to linear optimization". Athena Scientific, 1997. ISBN 978-1-886529-19-9. pp. 14, 75

- [30] BIEGEL Benjamin, ANDERSEN P, PEDERSEN TS et al. "Smart grid dispatch strategy for ON/OFF demand-side devices". In *European control conference (ECC)*. JUL 2013, pp. 2541–2548. ISBN 978-3-033-03962-9. p. 24
- [31] BIEGEL Benjamin, ANDERSEN P, STOUSTRUP J et al. "Lumped thermal household model". In *ISGT Europe*. OCT 2013. DOI 10.1109/ISGTEurope.2013.6695320. pp. 23, 24
- [32] BIEGEL Benjamin, ANDERSEN P, STOUSTRUP J & BENDTSEN JD. "Congestion management in a smart grid via shadow prices". In *8th IFAC symposium on power plant and power system control*. SEP 2012, pp. 518–523. DOI 10.3182/20120902-4-FR-2032.00091. p. 21
- [33] BLIEK Frits W, van den NOORT A, ROOSSEN B et al. "PowerMatching City, a living lab smart grid demonstration". In *ISGT Europe*. IEEE PES, OCT 2010. DOI 10.1109/ISGTEUROPE.2010.5638863. p. 271
- [34] BLOKKER Elisabeth JM. "Stochastic water demand modelling for a better understanding of hydraulics in water distribution networks". PhD thesis. Delft University of Technology, 2010. ISBN 978-90-8957-015-4. pp. 277, 278
- [35] BLUM Christian & ROLI A. "Metaheuristics in combinatorial optimization: overview and conceptual comparison". In *ACM computing surveys* 35 : 3, ACM, SEP 2003, pp. 268–308. ISSN 0360-0300. DOI 10.1145/937503.937505. pp. 15, 154
- [36] BOCCARD Nicolas. "Capacity factor of wind power realized values vs. estimates". In *Energy policy* 37 : 7, 2009, pp. 2679–2688. ISSN 0301-4215. DOI 10.1016/j.enpol.2009.02.046. p. 243
- [37] BONILLA Juan J, BLANCO JM, LÓPEZ L & SALA JM. "Technological recovery potential of waste heat in the industry of the Basque Country". In *Applied thermal engineering* 17 : 3, 1997, pp. 283–288. DOI 10.1016/S1359-4311(96)00024-5. p. 251
- [38] BORENSTEIN Severin. "Peak-time rebates: money for nothing?" MAY 2014. <http://www.greentechmedia.com/articles/read/Peak-Time-Rebates-Money-for-Nothing> (⌚ MAY 27, 2016). p. 17
- [39] BORGGREFE Frieder & NEUHOFF K. "Balancing and intraday market design: options for wind integration". In *DIW Berlin discussion paper*. 2011. <http://hdl.handle.net/10419/61319> (⌚ SEP 19, 2016). p. 11
- [40] BOSMAN Maurice GC. "Planning in smart grids". PhD thesis. Enschede: University of Twente, JUL 2012. ISBN 978-90-365-3386-7. DOI 10.3990/1.9789036533867. pp. 21, 32, 33, 42, 44, 45, 66, 69, 72, 74, 76, 77, 91, 131, 234, 252, 265
- [41] BOYD Stephen, XIAO L & MUTAPCIC A. "Subgradient methods". Lecture notes. OCT 2003. https://web.stanford.edu/class/ee392o/subgrad_method.pdf (⌚ MAY 31, 2015). p. 21
- [42] BREUKERS Sylvia C & MOURIK RM. "The end-users as starting point for designing dynamic pricing approaches to change household energy consumption behaviours". Tech. report. Report for Netbeheer Nederland, project group Smart Grids. Arnhem: DuneWorks, MAR 2013. http://www.duneworks.nl/wp-content/uploads/2014/02/2013-Netbeheer-Dynamic-Pricing-and-behaviour-change_Duneworks_2013.pdf (⌚ MAY 15, 2015). p. 17
- [43] BRITZ Wolfgang & DELZEIT R. "The impact of German biogas production on European and global agricultural markets, land use and the environment". In *Energy policy* 62, 2013, pp. 1268–1275. ISSN 0301-4215. DOI 10.1016/j.enpol.2013.06.123. p. 243
- [44] BSW-SOLAR & SOLAR PROMOTION. "Price drop for solar power storage systems". Press release. DEC 2014. <http://www.solarwirtschaft.de/en/media/single-view/news/price-drop-for-solar-power-storage-systems.html> (⌚ MAY 10, 2015). p. 266

- [45] BUENO C. & CARTA JA. "Wind powered pumped hydro storage systems, a means of increasing the penetration of renewable energy in the Canary Islands". In *Renewable and sustainable energy reviews* 10: 4, 2006, pp. 312–340. ISSN 1364-0321. DOI 10.1016/j.rser.2004.09.005. p. 244
- [46] BUI Nicola, CASTELLANI AP, CASARI P & ZORZI M. "The internet of energy: a web-enabled smart grid system". In *Network, IEEE* 26: 4, JUL 2012, pp. 39–45. ISSN 0890-8044. DOI 10.1109/MNET.2012.6246751. p. 260
- [47] BUNDESMINISTERIUM FÜR WIRTSCHAFT UND ENERGIE. "SmartHome2Market: Marktperspektiven für die intelligente Heimvernetzung". *German*. JUN 2016. <http://www.digitale-technologien.de/DT/Redaktion/DE/Downloads/Publikation/smarthome-broschuere.pdf> (⌚ JUL 3, 2016). p. 193
- [48] BUNDESNETZAGENTUR. "Degressions- und Vergütungssätze bis Dezember 2014". *German*. http://www.bundesnetzagentur.de/cln_1432/DE/Sachgebiete/ElektrizitaetundGas/Unternehmen_Institutionen/ErneuerbareEnergien/Photovoltaik/ArchivDatenMeldgn/ArchivDatenMeldgn_node.html (⌚ MAR 15, 2015). p. 240
- [49] BURGER Bruno. "Electricity production from solar and wind in Germany in 2014". Fraunhofer institute for solar energy systems (ISE), 2014. <http://www.ise.fraunhofer.de/en/downloads-englisch/pdf-files-englisch/data-nivc/electricity-production-from-solar-and-wind-in-germany-2014.pdf> (⌚ MAR 30, 2015). p. 240
- [50] BURKE Edmund K, GENDREAU M, HYDE M et al. "Hyper-heuristics: a survey of the state of the art". In *Journal of the operational research society* 64: 12, Palgrave Macmillan, DEC 2013, pp. 1695–1724. ISSN 0160-5682. DOI 10.1057/jors.2013.71. p. 154
- [51] BUŞONIU Lucian, BABUŞKA R, DE SCHUTTER B & ERNST D. "Reinforcement learning and dynamic programming using function approximators". *Automation and Control Engineering Series*. Taylor & Francis Group, 2010. ISBN 978-1-4398-2108-4. pp. 203, 209
- [52] BUTTAZZO Giorgio C. "Hard real-time computing systems: predictable scheduling algorithms and applications". 3rd edition. Vol. 24. Real-time systems series. Springer, 2011. ISBN 978-1-4614-0675-4. DOI 10.1007/978-1-4614-0676-1. p. 17
- [53] CALIFORNIA ENERGY COMMISSION. "2013 building energy efficiency standards". Web site. MAY 2012. <http://www.energy.ca.gov/title24/2013standards/> (⌚ JUN 6, 2015). p. 263
- [54] CALIFORNIA INDEPENDENT SYSTEM OPERATOR. "What the duck curve tells us about managing a green grid". Fact sheet. OCT 2013. http://www.aiso.com/Documents/FlexibleResourcesHelpRenewables_FastFacts.pdf (⌚ JUN 9, 2015). p. 242
- [55] CALIFORNIA LIGHTING TECHNOLOGY CENTER. "What's new in the 2013 code? Changes to mandatory Title 24 lighting requirements". Guide. MAR 2014. <http://cltc.ucdavis.edu/sites/default/files/files/publication/title-24-2013-code-changes-140318.pdf> (⌚ JUN 6, 2015). p. 263
- [56] CARØE Claus C & TIND J. "L-shaped decomposition of two-stage stochastic programs with integer recourse". In *Mathematical programming* 83: 1–3, Springer, 1998, pp. 451–464. ISSN 0025-5610. DOI 10.1007/BF02680570. p. 22
- [57] CHASSIN David P, SCHNEIDER K & GERKENSMEYER C. "GridLAB-D: an open-source power systems modeling and simulation environment". In *Transmission and distribution (T&D) conference and exposition*. IEEE PES, APR 2008. DOI 10.1109/TDC.2008.4517260. p. 30
- [58] CHOU Andy, YANG J, CHELF B et al. "An empirical study of operating systems errors". In *SIGOPS operating systems review* 35: 5, ACM, OCT 2001, pp. 73–88. ISSN 0163-5980. DOI 10.1145/502059.502042. p. 192

- [59] CLAESSEN Felix N. "Smart grid control: an analysis of control systems within a Dutch residential microgrid incorporating decentralised renewable energy resources". MSc thesis. Utrecht University, DEC 2011. pp. 273, 277, 278
- [60] CLAESSEN Felix N, HÖNING NF, LIEFERS BJ et al. "Market Garden: a scalable research environment for heterogeneous electricity markets". In *ERCIM news* '92, ERCIM, JAN 2013, pp. 25–26. ISSN 0926-4981. <http://oai.cwi.nl/oai/asset/22074/22074D.pdf> (⌚ MAY 17, 2015). p. 30
- [61] CLAESSENS Bert J, VANDAELE S, RUELENS F et al. "Peak shaving of a heterogeneous cluster of residential flexibility carriers using reinforcement learning". In *ISGT Europe*. IEEE PES, OCT 2013. DOI 10.1109/ISGTEurope.2013.6695254. p. 23
- [62] CLAESSENS Bert J, VANDAELE S, RUELENS F & HOMMELBERG MPF. "Self-learning demand side management for a heterogeneous cluster of devices with binary control actions". In *ISGT Europe*. IEEE PES, OCT 2012. DOI 10.1109/ISGTEurope.2012.6465679. pp. 23, 37
- [63] CLOÏN Mariëlle, KAMPHUIS C, SCHOLS M et al. "Nederland in een dag: tijdsbesteding in Nederland vergeleken met die in 15 andere Europese landen". *Dutch*. Sociaal en Cultureel Planbureau, OCT 2011. http://www.scp.nl/Publicaties/Alle_publicaties/Publicaties_2011/Nederland_in_een_dag (⌚ MAY 26, 2016). p. 277
- [64] COCHRAN Jaquelin, LEW D & KUMAR N. "Flexible coal: evolution from baseload to peaking plant". Tech. report NREL/BR-6A20-60575. National Renewable Energy Laboratory, DEC 2013. http://www.power-eng.com/blogs/power-points/2014/03/the_evolution_coalpl.html (⌚ MAY 29, 2015). p. 241
- [65] CONTRERAS Javier, CANDILES O, de la FUENTE JI & GÓMEZ T. "Auction design in day-ahead electricity markets". In *Power systems, IEEE transactions on* '16: 3, AUG 2001, pp. 409–417. ISSN 0885-8950. DOI 10.1109/59.932276. p. 13
- [66] CRAWLEY Drury B, LAWRIE LK, WINKELMANN FC et al. "EnergyPlus: creating a new-generation building energy simulation program". In *Energy and buildings* '33: 4, 2001, pp. 319–331. ISSN 0378-7788. DOI 10.1016/S0378-7788(00)00114-6. p. 30
- [67] CURTRIGHT Aimee E & APT J. "The character of power output from utility-scale photovoltaic systems". In *Progress in photovoltaics: research and applications* '16, MAY 2008, pp. 241–247. DOI 10.1002/pip.786. p. 242
- [68] DAGIOGLOU Stergios. "An implementation of the planning phase of TRIANA using the Flexible Power Application Infrastructure". MSc thesis. University of Twente, AUG 2014. <http://purl.utwente.nl/essays/70985> (⌚ SEP 19, 2016). pp. 191, 192
- [69] DALL'ANESE Emiliano, ZHU H & GIANNAKIS GB. "Distributed optimal power flow for smart microgrids". In *Smart grid, IEEE transactions on* '4: 3, SEP 2013, pp. 1464–1475. ISSN 1949-3053. DOI 10.1109/TSG.2013.2248175. p. 21
- [70] DANTZIG George B & INFANGER G. "Multi-stage stochastic linear programs for portfolio optimization". In *Annals of operations research* '45: 1, DEC 1993, pp. 59–76. ISSN 0254-5330. DOI 10.1007/BF02282041. p. 16
- [71] DANTZIG George B & WOLFE P. "The decomposition algorithm for linear programs". In *Econometrica* '29: 4, The Econometric Society, 1961, pp. 767–778. ISSN 0012-9682. DOI 10.2307/1911818. pp. 21, 80
- [72] DANTZIG George B. "Linear programming and extensions". R-366-PR. Princeton University Press, AUG 1963. <http://www.rand.org/pubs/reports/R366.html> (⌚ SEP 19, 2016). pp. 76, 78, 91
- [73] DE BRABANDERE Karel. "Voltage and frequency droop control in low voltage grids by distributed generators with inverter front-end". PhD thesis. Katholieke Universiteit Leuven, OCT 2006. ISBN 978-90-5682-745-8. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.126.3675&rep=rep1&type=pdf> (⌚ AUG 28, 2016). p. 18

- [74] DE CRAEMER Klaas. “Event-driven demand response for electric vehicles in multi-aggregator distribution grid settings”. PhD thesis. KU Leuven, JUL 2014. ISBN 978-94-6018-862-6. https://lirias.kuleuven.be/bitstream/123456789/454402/1/ThesisKlaasDeCraemer_July2014.pdf (⌚ SEP 19, 2016). p. 19, 22–24, 28
- [75] DE CRAEMER Klaas, VANDAELE S, CLAESSENS BJ & DECONINCK G. “Integration of distribution grid constraints in an event-driven control strategy for plug-in electric vehicles in a multi-aggregator setting”. In *Plug in electric vehicles in smart grids*. Edited by RAJAKARUNA S, SHAHNIA F & GHOSH A. Power Systems. Springer, 2015, pp. 129–171. ISBN 978-981-287-301-9. DOI 10.1007/978-981-287-302-6_6. p. 19
- [76] DEUTSCHE ENERGIE-AGENTUR. “Ausbau- und Innovationsbedarf der Stromverteilnetze in Deutschland bis 2030”. German. DEC 2012. http://www.dena.de/fileadmin/user_upload/Projekte/Energiesysteme/Dokumente/denaVNS_Abschlussbericht.pdf (⌚ MAY 26, 2016). p. 282
- [77] DEUTSCHE WELLE. “German power net survives solar eclipse”. News article. MAR 2015. <http://dw.de/p/1EumM> (⌚ MAY 11, 2015). p. 267
- [78] DIVYA Kurthakoti C & ØSTERGAARD J. “Battery energy storage technology for power systems – an overview”. In *Electric power systems research* 79 : 4, 2009, pp. 511–520. ISSN 0378-7796. DOI 10.1016/j.epsr.2008.09.017. pp. 258, 270
- [79] DNV GL & CE DELFT. “System integration – hybrid energy infrastructures”. GCS.103750. <http://topsectorenergie.nl/wp-content/uploads/2015/04/Systeemintegratie-Eindrapport-Perceel-3-Hybride-Infrastructuren-DNVGL-22-april-2015.pdf> (⌚ JUL 11, 2016). p. 250
- [80] DNV GL, IMPERIAL COLLEGE & NERA ECONOMIC CONSULTING. “Integration of renewable energy in Europe”. Tech. report 9011-700. JUN 2014. https://ec.europa.eu/energy/sites/ener/files/documents/201406_report_renewables_integration_europe.pdf (⌚ JUL 4, 2016). pp. 1, 2
- [81] DONG Leilei, LIU H & RIFFAT S. “Development of small-scale and micro-scale biomass-fuelled CHP systems – a literature review”. In *Applied thermal engineering* 29 : 11–12, 2009, pp. 2119–2126. ISSN 1359-4311. DOI 10.1016/j.applthermaleng.2008.12.004. pp. 251, 252
- [82] DOORMAN Gerard L & GRANDE OS. “Reserve requirements and price spikes in multinational power markets”. In *7th international conference on the European energy market (EEM)*. JUN 2010. DOI 10.1109/EEM.2010.5558731. p. 244
- [83] DÖRING Michael, BURGESS K, HOFMANN F et al. “Development of a retrofit program for distributed generation in Germany for the prevention of frequency stability problems in abnormal system conditions”. Tech. report POWDE14322. Ecofys, JUN 2014. <http://www.ecofys.com/files/files/ecofys-wg-bbh-ifk-2014-retrofit-program-distributed-generation.pdf> (⌚ MAY 16, 2015). p. 18
- [84] DOUGLASS Dale A & EDRIS A. “Real-time monitoring and dynamic thermal rating of power transmission circuits”. In *Power delivery, IEEE transactions on* 11 : 3, JUL 1996, pp. 1407–1418. ISSN 0885-8977. DOI 10.1109/61.517499. p. 258
- [85] van DUIN Coen & STOELDRAIJER L. “Huishoudensprognose 2011–2060: meer en kleinere huishoudens”. Dutch. Centraal Bureau voor de Statistiek, 2011. <https://www.cbs.nl/nl-nl/achtergrond/2011/27/huishoudensprognose-2011-2060-meer-en-kleinere-huishoudens> (⌚ MAY 26, 2016). p. 278
- [86] DUNN Bruce, KAMATH H & TARASCON J. “Electrical energy storage for the grid: a battery of choices”. In *Science* 334 : 6058, 2011, pp. 928–935. DOI 10.1126/science.1212741. p. 258
- [87] E-BALANCE. “Balancing energy production and consumption in energy efficient smart neighbourhoods”. FP7 research project. 2013–2017. <http://www.e-balance-project.eu/> (⌚ MAY 25, 2016). p. 227

- [88] ECKSTEIN Jonathan & YAO W. “Augmented Lagrangian and alternating direction methods for convex optimization: a tutorial and some illustrative computational results”. Tech. report RRR 32-2012. DEC 2012. http://rutcor.rutgers.edu/pub/rrr/reports2012/32_2012.pdf (⌚ JUN 1, 2015). p. 21
- [89] ECN, ENERGIE-NEDERLAND & NETBEHEER NEDERLAND. “Energietrends 2014”. *Dutch*. White paper. SEP 2014. <https://issuu.com/aandagt/docs/energietrends2014> (⌚ AUG 28, 2016). pp. 13, 240, 241, 244, 245, 254
- [90] ELEMENT ENERGY. “Cost and performance of EV batteries”. Tech. report. The Committee on Climate Change, MAR 2012. http://www.element-energy.co.uk/wordpress/wp-content/uploads/2012/04/CCC-battery-cost_-Element-Energy-report_March2012_Public.pdf (⌚ APR 7, 2015). p. 247
- [91] ELEMENT ENERGY. “Pathways to high penetration of electric vehicles”. Tech. report. The Committee on Climate Change, DEC 2013. http://www.theccc.org.uk/wp-content/uploads/2013/12/CCC-EV-pathways_FINAL-REPORT_17-12-13-Final.pdf (⌚ APR 8, 2015). p. 248
- [92] ELMER Theo, WORALL M, WU S & RIFFAT SB. “Emission and economic performance assessment of a solid oxide fuel cell micro-combined heat and power system in a domestic building”. In *Applied thermal engineering* 90 : 5, Nov 2015, pp. 1082–1089. ISSN 1359-4311. DOI 10.1016/j.applthermaleng.2015.03.078. p. 252
- [93] ELTAWIL Mohamed A & ZHAO Z. “Grid-connected photovoltaic power systems: technical and potential problems – a review”. In *Renewable and sustainable energy reviews* 14 : 1, Elsevier, 2010, pp. 112–129. DOI 10.1016/j.rser.2009.07.015. p. 242
- [94] ENECO. “Henk charges his Tesla when electricity price is low: Eneco develops new app in response to increase in sustainable electricity”. Press release. MAR 2015. <http://news.eneco.com/henk-charges-his-tesla-when-electricity-price-is-low/> (⌚ JUN 6, 2015). p. 264
- [95] ENECO. “Toon thermostaat”. *Dutch*. Web site. <https://www.eneco.nl/toon-thermostaat/> (⌚ JUN 6, 2015). p. 264
- [96] ENERGIEGIDS. “Netbeheerders: integrale aanpak duurzame renovatie”. *Dutch*. Article. MAR 2015. http://www.energiegids.nl/sites/default/files/archief/a37f7e/eg0102_2015_pag.24-27.pdf (⌚ JUN 10, 2015). p. 242
- [97] ENERGIEKAART. “Smart grid: rendement voor iedereen”. *Dutch*. 2011–2015. <https://www.energiekaart.net/initiatieven/smart-grid-rendement-voor-iedereen/> (⌚ MAY 27, 2016). p. 293
- [98] ENERGY@HOME. “Energy@home: a ‘user-centric’ energy management system”. White paper. MAY 2011. http://www.energy-home.it/Documents/Others/110518Energy@Home_Whitepaper.pdf (⌚ JUN 6, 2015). p. 264
- [99] ENERNOC. “Demand response: a multi-purpose resource for utilities and grid operators”. White paper. JUL 2009. http://www.enernoc.com/themes/bluemasters/images/brochures/pdfs/2-Whitepaper-DR-A_Multi-Purpose_Resource.pdf (⌚ JUN 6, 2015). p. 265
- [100] ENERNOC. “EnerNOC demand response turns your back-up generator into a revenue engine”. Product brochure. JUN 2014. http://www.enernoc.com/themes/bluemasters/images/brochures/pdfs/br_generators.pdf (⌚ JUN 6, 2015). p. 265
- [101] ENEXIS. “Elektrisch rijden”. *Dutch*. Web site. <https://www.enexisinnovatie.nl/themas/elektrisch-rijden/> (⌚ AUG 27, 2016). p. 248
- [102] ENGEL Marijke. “Costly ruins of the energy transition”. Magazine article. Original link no longer available, mirrored at (http://web.archive.org/web/*/http://www.whome.cs.utwente.nl/~toerscheha/phd/mirror/GaswinnerCostlyRuins.pdf). Gaswinner. <https://www.wingas.com/en/media-library/gaswinner/archive/costly-ruins-of-the-energy-transition.html> (⌚ MAR 31, 2015). p. 241

- [103] ENIPEDIA. “Tesla Model S battery”. Delft University of Technology, Nov 2014. http://enipedia.tudelft.nl/wiki/Tesla_Model_S_Battery (⌚ APR 7, 2015). pp. 247, 248
- [104] ENTSO-E. “Solar eclipse 2015 impact analysis”. Tech. report. FEB 2015. https://www.entsoe.eu/Documents/Publications/SOC/150219_Solar_Eclipse_Impact_Analysis_Final.pdf (⌚ MAY 17, 2015). p. 267
- [105] ENTSO-E. “Survey on ancillary services procurement & balancing market design”. Presentation. SEP 2012. https://www.entsoe.eu/fileadmin/user_upload/_library/resources/BAL/121022_Survey_on_AS_Procurement_and_EBM_design.pdf (⌚ MAY 12, 2015). p. 12
- [106] E.ON. “2014 annual report”. MAR 2015. <http://www.eon.com/en/about-us/publications/annual-report.html> (⌚ MAR 31, 2015). p. 241
- [107] EUPD RESEARCH. “2nd Storage & Solar Briefing: Experten im Gespräch, Interviews und Markteinschätzungen”. *German*. White paper. 2013. http://eupd-research.com/fileadmin/templates/images/mailer/Storage/Storage_Interviews.pdf (⌚ MAY 10, 2015). p. 266
- [108] EUROPEAN COMMISSION. “2030 framework for climate and energy policies”. Original link no longer available, mirrored at http://web.archive.org/web/20150707210603/http://ec.europa.eu/clima/policies/2030/index_en.htm, see also <http://ec.europa.eu/energy/en/topics/energy-strategy/2030-energy-strategy>. OCT 2014. http://ec.europa.eu/clima/policies/2030/index_en.htm (⌚ MAY 28, 2015). p. 240
- [109] EUROPEAN COMMISSION. “A roadmap for moving to a competitive low carbon economy in 2050”. MAR 2011. http://eur-lex.europa.eu/resource.html?uri=cellar:5db26ecc-ba4e-4de2-ae08-dba649109d1d.0002.03/DOC_1&format=PDF (⌚ MAY 28, 2015). p. 240
- [110] EUROPEAN FORUM FOR RENEWABLE ENERGY SOURCES. “EU tracking roadmap 2014: keeping track of renewable energy targets towards 2020”. Report. JUN 2014. http://www.keepontrack.eu/contents/publicationseutrackingroadmap/kot_eutrackingroadmap2014.pdf (⌚ MAR 6, 2015). p. 240
- [111] EUROPEAN PARLIAMENT & COUNCIL OF THE EUROPEAN UNION. “Decision No 406/2009/EC of the European Parliament and of the council of 23 April 2009 on the efforts of member states to reduce their greenhouse gas emissions to meet the community’s emission reduction commitments up to 2020”. In *Official journal of the European Union*, APR 2009, pp. 136–148. p. 240
- [112] FADLULLAH Zubair M, NOZAKI Y, TAKEUCHI A & KATO N. “A survey of game theoretic approaches in smart grid”. In *Wireless communications and signal processing (WCSP), 2011 international conference on*. Nov 2011. DOI 10.1109/WCSP.2011.6096962. p. 30
- [113] FAIRLEY Peter. “Norway wants to be Europe’s battery”. IEEE Spectrum. OCT 2014. <http://spectrum.ieee.org/green-tech/wind/norway-wants-to-be-europes-battery> (⌚ MAR 31, 2015). p. 242
- [114] FARHANGI Hassan. “The path of the smart grid”. In *IEEE power and energy magazine* 8:1, 2010, pp. 18–28. pp. 258, 259
- [115] FARREL Adrian. “IANA considerations for Three Letter Acronyms”. RFC 5513. APR 2009. <https://www.ietf.org/rfc/rfc5513.txt> (⌚ SEP 18, 2014). p. 347
- [116] FARUQUI Ahmad, HARRIS D & HLEDIK R. “Unlocking the €53 billion savings from smart meters in the EU: how increasing the adoption of dynamic tariffs could make or break the EU’s smart grid investment”. In *Energy policy* 38:10, 2010, pp. 6222–6231. ISSN 0301-4215. DOI 10.1016/j.enpol.2010.06.010. p. 17
- [117] FLEXIBLEPOWER ALLIANCE NETWORK. “Energy Flexibility Platform & Interface documentation”. <http://fpai-ci.sensorlab.tno.nl/builds/fpai-documentation/development/html/> (⌚ MAY 23, 2016). p. 197

- [118] FLEXIBLEPOWER ALLIANCE NETWORK. "GitHub source repository". <https://github.com/flexiblepower/> (🔗 MAR 12, 2015). p. 195
- [119] FLEXIBLEPOWER ALLIANCE NETWORK. "PowerMatcher-EF-Pi BufferBid implementation". GitHub source repository. MAY 2015. <https://github.com/flexiblepower/fpai-apps/blob/master/net.powermatcher.fpai.controller/src/net/powermatcher/fpai/agents/BufferBid.java> (🔗 MAR 4, 2016). p. 166
- [120] FLEXITRICITY. "Flexitricity: unlocking smart grid revenue". Brochure. Original link no longer available, mirrored at (http://web.archive.org/web/*/http://www.whome.cs.utwente.nl/~toerscheha/phd/mirror/FlexitricityBrochure.pdf), latest version at <https://www.flexitricity.com/en-gb/a/knowledge-centre/brochures/service-brochure/>. AUG 2011. <http://www.flexitricity.com/docLibrary/Flexitricity%20Brochure%201.0.pdf> (🔗 JUN 6, 2015). p. 265
- [121] FONG Square KF, HANBY VI & CHOW TT. "HVAC system optimization for energy management by evolutionary programming". In *Energy and buildings* 38 : 3, 2006, pp. 220–231. ISSN 0378-7788. DOI 10.1016/j.enbuild.2005.05.008. p. 22
- [122] FRONTIER ECONOMICS. "Qualitative assessment of balancing market design for the Netherlands, a report prepared for Energiekamer". Tech. report. APR 2011. <https://www.acm.nl/download/documenten/nma/103542%20Rapport.pdf> (🔗 MAY 11, 2015). pp. 11, 12
- [123] FRUNT Jasper. "Analysis of balancing requirements in future sustainable and reliable power systems". PhD thesis. Technische Universiteit Eindhoven, JUN 2011. ISBN 978-90-386-2484-6. DOI 10.6100/IR711271. pp. 12, 259
- [124] FULLER Jason C, SCHNEIDER KP & CHASSIN DP. "Analysis of residential demand response and double-auction markets". In *Power and energy society general meeting (PES GM)*. IEEE PES, JUL 2011. DOI 10.1109/PES.2011.6039827. p. 24
- [125] GEELEN Daphne, REINDERS A & KEYSON D. "Empowering the end-user in smart grids: recommendations for the design of products and services". In *Energy policy* 61, 2013, pp. 151–161. ISSN 0301-4215. DOI 10.1016/j.enpol.2013.05.107. p. 264
- [126] GELLINGS Clark. "Estimating the costs and benefits of the smart grid: a preliminary estimate of the investment requirements and the resultant benefits of a fully functioning smart grid". Tech. report 1022519. Electric Power Research Institute, MAR 2011. p. 1
- [127] GEOFFRION Arthur M. "Generalized Benders decomposition". In *Journal of optimization theory and applications* 10 : 4, Springer, OCT 1972, pp. 237–260. ISSN 0022-3239. DOI 10.1007/BF00934810. p. 22
- [128] GERARDS Marco ET, TOERSCHE HA, HOOGSTEEN G et al. "Visualization of profile steering algorithm". <http://www.whome.ewi.utwente.nl/~gerardsmet/vis/ps/> (🔗 MAY 14, 2016). p. 107
- [129] GHASEMIEH Hamed, HAVERKORT BRHM, JONGERDEN MR & REMKE AKI. "Energy resilience modeling for smart houses". In *Proceedings of the 45th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*. Rio de Janeiro, Brazil: IEEE Computer Society, JUN 2015, pp. 275–286. ISBN 978-1-4799-8629-3. DOI 10.1109/DSN.2015.31. p. 10
- [130] GHIJSEN Mattijs & D'HULST R. "Market-based coordinated charging of electric vehicles on the low-voltage distribution grid". In *Smart grid modeling and simulation (SGMS), 2011 IEEE first international workshop on*. 2011. DOI 10.1109/SGMS.2011.6089021. p. 27
- [131] GILMORE Paul C & GOMORY RE. "A linear programming approach to the cutting stock problem—Part II". In *Operations research* 11 : 6, INFORMS, 1963, pp. 863–888. ISSN 0030-364X. DOI 10.1287/opre.11.6.863. p. 78

- [132] GILMORE Paul C & GOMORY RE. “A linear programming approach to the cutting-stock problem”. In *Operations research* ⁹: 6, INFORMS, 1961, pp. 849–859. ISSN 0030-364x. DOI 10.1287/opre.9.6.849. p. 78
- [133] GISELSSON Pontus, DOAN MD, KEVICZKY T et al. “Accelerated gradient methods and dual decomposition in distributed model predictive control”. In *Automatica* ⁴⁹: 3, 2013, pp. 829–833. ISSN 0005-1098. DOI 10.1016/j.automatica.2013.01.009. p. 21
- [134] GLASSEY C Roger. “Nested decomposition and multi-stage linear programs”. In *Management science* ²⁰: 3, INFORMS, 1973, pp. 282–292. ISSN 0025-1909. DOI 10.1287/mnsc.20.3.282. p. 80
- [135] GÓMEZ OÑATE Virginia, KESSELS K & SIX D. “Best-practices for district heating implementation in Belgium: policy analysis and benchmark from countries with high market penetration”. In *EnergyCon*. IEEE, MAY 2014, pp. 1083–1089. ISBN 978-1-4799-2448-6. DOI 10.1109/ENERGYCON.2014.6850551. p. 251
- [136] GONDZIO Jacek, GONZÁLEZ BREVIS P & MUNARI P. “Large-scale optimization with the primal-dual column generation method”. In *Mathematical programming computation* ⁸: 1, 2016, pp. 47–82. ISSN 1867-2957. DOI 10.1007/s12532-015-0090-6. p. 106
- [137] GONZÁLEZ BREVIS Pablo. “Advances in interior point methods and column generation”. PhD thesis. University of Edinburgh, 2013. <https://www.era.lib.ed.ac.uk/bitstream/handle/1842/8052/Gonz%C3%83%C2%A1lez%20Brevis2013.pdf?sequence=2&isAllowed=y> (⌚ JUN 13, 2016). p. 106
- [138] GOODMAN. “DSXC18 split system air conditioner, up to 18 SEER”. Product brochure. <http://www.goodmanmfg.com/Portals/0/pdf/SS/SS-DSXC18.pdf> (⌚ APR 7, 2015). p. 247
- [139] GOOGLE. “Protocol buffers”. <https://developers.google.com/protocol-buffers/> (⌚ OCT 7, 2014). pp. 202, 316
- [140] GREEN Robert C, WANG L & ALAM M. “The impact of plug-in hybrid electric vehicles on distribution networks: a review and outlook”. In *Renewable and sustainable energy reviews* ¹⁵: 1, 2011, pp. 544–553. ISSN 1364-0321. DOI 10.1016/j.rser.2010.08.015. p. 248
- [141] GRIDWISE ARCHITECTURE COUNCIL. “GridWise transactive energy framework, version 1.0”. Tech. report PNNL-22946. JAN 2015. http://www.gridwiseac.org/pdfs/te_framework_report_pnnl-22946.pdf (⌚ MAY 21, 2015). p. 17
- [142] GROWE-KUSKA Nicole, HEITSCH H & ROMISCH W. “Scenario reduction and scenario tree construction for power management problems”. In *PowerTech*. Bologna: IEEE PES, JUN 2003. DOI 10.1109/PTC.2003.1304379. p. 15
- [143] GUSTAVSSON Rune. “Agents with power”. In *Communications of the ACM* ⁴²: 3, ACM, MAR 1999, pp. 41–47. ISSN 0001-0782. DOI 10.1145/295685.295698. p. 30
- [144] HAHN Henning, KRAUTKREMER B, HARTMANN K & WACHENDORF M. “Review of concepts for a demand-driven biogas supply for flexible power generation”. In *Renewable and sustainable energy reviews* ²⁹, 2014, pp. 383–393. ISSN 1364-0321. DOI 10.1016/j.rser.2013.08.085. p. 243
- [145] HALVGAARD Rasmus, JORGENSEN JB, POULSEN NK et al. “Decentralized large-scale power balancing”. In *ISGT Europe*. IEEE PES, OCT 2013. DOI 10.1109/ISGTEurope.2013.6695323. p. 21
- [146] HAMMERSTROM Donald J, AMBROSIO R, CARLON TA et al. “Pacific Northwest GridWise testbed demonstration projects, part I: Olympic Peninsula project”. Tech. report PNNL-17167. Pacific Northwest National Laboratory, OCT 2007. p. 270
- [147] HAMMERSTROM Donald J, BROUS J, CHASSIN DP et al. “Pacific Northwest GridWise testbed demonstration projects, part II: Grid Friendly Appliance project”. Tech. report PNNL-17079. Pacific Northwest National Laboratory, OCT 2007. pp. 18, 265, 269, 270

- [148] HANEY Aoife B, JAMASB T & POLLITT M. "Smart metering and electricity demand: technology, economics and international experience". Working paper. FEB 2009. <http://www.econ.cam.ac.uk/research/repec/cam/pdf/cwpe0905.pdf> (⌚ JUN 8, 2015). p. 18
- [149] HARVEY Scott M & HOGAN WW. "Nodal and zonal congestion management and the exercise of market power". JAN 2000. http://www.hks.harvard.edu/fs/whogan/zonal_jan10.pdf (⌚ MAY 13, 2015). p. 15
- [150] HAVEL Petr & MALÍK O. "Optimal dispatch of regulation reserves for power balance control in transmission system". In *International conference on control and automation*. IEEE. 2009, pp. 807–812. p. 13
- [151] HELLEMA Duco, WIEBES C & WITTE T. "The Netherlands and the oil crisis: business as usual". Translated by PEARSON M. Amsterdam University Press, 2004. ISBN 978-90-5356-485-1. p. 239
- [152] HEUSSEN Kai, YOU S, BIEGEL B et al. "Indirect control for demand side management – a conceptual introduction". In *ISGT Europe*. IEEE PES, OCT 2012, pp. 1–8. DOI 10.1109/ISGTEurope.2012.6465858. p. 18
- [153] HILDMANN Marcus, ULBIG A & ANDERSSON G. "Revisiting the merit-order effect of renewable energy sources". Tech. report. arXiv, JUL 2013. <http://EconPapers.repec.org/RePEc:arx:papers:1307.0444> (⌚ SEP 25, 2016). p. 265
- [154] HOBBS Benjamin F. "Optimization methods for electric utility resource planning". In *European journal of operational research* 83:1, 1995, pp. 1–20. ISSN 0377-2217. DOI 10.1016/0377-2217(94)00190-N. p. 237
- [155] HOLLINGER Raphael, WILLE-HAUSSMANN B, ERGE T et al. "Speicherstudie 2013: Kurzgutachten zur Abschätzung und Einordnung energiewirtschaftlicher, ökonomischer und anderer Effekte bei Förderung von objektgebunden elektrochemischen Speichern, Zusammenfassung der wichtigsten Erkenntnisse". *German*. Tech. report. Fraunhofer institute for solar energy systems (ISE), JAN 2013. <http://www.ise.fraunhofer.de/de/veroeffentlichungen/veroeffentlichungen-pdf-dateien/studien-und-konzeptpapiere/speicherstudie-2013.pdf> (⌚ JUN 6, 2015). pp. 266, 270
- [156] HOMMELBERG Maarten PF & CLAESSENS BJ. "System for distributing electrical energy over a cluster of electrical devices, method for distributing electrical energy over a cluster of electrical devices of such a system and controller for use in such a system". PCT/EP2012/076882. JUN 2013. <http://www.google.com/patents/WO2013093119A1> (⌚ JUN 3, 2015). p. 23
- [157] HÖNING Nicolas., DE JONG E, BLOEMHOF G & LA POUTRÉ H. "Thermal behaviour of low voltage cables in smart grid-related environments". In *ISGT Europe*. Istanbul, Turkey: IEEE PES, OCT 2014. ISBN 978-1-4799-7720-8. DOI 10.1109/ISGTEurope.2014.7028736. pp. 257, 258
- [158] HÖNING Nicolas & LA POUTRÉ H. "Reduction of market power and stabilisation of outcomes in a novel and simplified two-settlement electricity market". In *Web intelligence and intelligent agent technology (WI-IAT), 2012 IEEE/WIC/ACM international conferences on*. DEC 2012, pp. 103–110. DOI 10.1109/WI-IAT.2012.168. p. 30
- [159] HOOGSTEEN Gerwin, MOLDERINK A, BAKKER V & SMIT GJM. "Integrating LV network models and load-flow calculations into smart grid planning". In *ISGT Europe*. Innovative Smart Grid Technologies Europe. Copenhagen, Denmark: IEEE PES, OCT 2013. DOI 10.1109/ISGTEurope.2013.6695427. pp. 33, 71, 75, 105
- [160] HOOGSTEEN Gerwin, MOLDERINK A, HURINK JL & SMIT GJM. "Managing energy in time and space in smart grids using TRIANA". In *ISGT Europe*. Istanbul, Turkey: IEEE PES, OCT 2014. DOI 10.1109/ISGTEurope.2014.7028973. pp. 34, 65

- [161] HOOGSTEEN Gerwin, MOLDERINK A, HURINK JL et al. "Impact of peak electricity demand in distribution grids: a stress test". In *PowerTech*. Eindhoven: IEEE PES, JUN 2015. DOI 10.1109/PTC.2015.7232412. p. 248
- [162] HOPPER Nicole, GOLDMAN C, GILLIGAN D et al. "A survey of the U.S. ESCO industry: market growth and development from 2000 to 2006". Tech. report LBNL-62679. Lawrence Berkeley National Laboratory, MAY 2007. <https://escholarship.org/uc/item/7wr0d765> (⌚ MAY 12, 2015). p. 12
- [163] van HOUT Marit, KOUTSTAAL P, OZDEMIR O & SEEBREGTS A. "Quantifying flexibility markets". Tech. report ECN-E-14-039. ECN, DEC 2014. <https://www.ecn.nl/publicaties/ECN-E-14-039> (⌚ SEP 19, 2016). p. 12
- [164] HUFEN Hans JAM & KOPPENJAN JFM. "Local renewable energy cooperatives: revolution in disguise?" In *Energy, sustainability and society* v5: 1, Springer, 2015. DOI 10.1186/s13705-015-0046-8. p. 249
- [165] IACOVELLA Sandro, LEMKENS K, GETH F et al. "Distributed voltage control mechanism in low-voltage distribution grid field test". In *ISGT Europe*. IEEE PES, OCT 2013. ISBN 978-1-4799-2984-9. DOI 10.1109/ISGTEurope.2013.6695393. p. 23
- [166] IACOVELLA Sandro, RUELENS F, VINGERHOETS P et al. "Cluster control of heterogeneous thermostatically controlled loads using tracer devices". In *Smart grid, IEEE transactions on* vPP: 99, IEEE PES, 2015. ISSN 1949-3053. DOI 10.1109/TSG.2015.2483506. pp. 28, 134, 187
- [167] IBM. "CPLEX performance tuning for mixed integer programs". FAQ article. DEC 2005. <http://www-01.ibm.com/support/docview.wss?uid=swg21400023> (⌚ MAY 31, 2015). pp. 14, 37
- [168] IEEE POWER AND ENERGY SOCIETY. "Piecing the picture together: natural gas & electricity". Magazine. Nov/DEC 2014. DOI 10.1109/MPE.2014.2340772. pp. 9, 241
- [169] INTERNATIONAL ENERGY AGENCY. "Implementing agreement on demand-side management technologies and programmes". Annual report 2014. JAN 2015. <http://www.ieadsm.org/wp/files/Annual-Report-2014.pdf> (⌚ AUG 28, 2016). p. 262
- [170] JABR Rabih A, COONICK AH & CORY BJ. "A homogeneous linear programming algorithm for the security constrained economic dispatch problem". In *Power systems, IEEE transactions on* v15: 3, AUG 2000, pp. 930–936. ISSN 0885-8950. DOI 10.1109/59.871715. p. 10
- [171] JENSEN Steen K. "Styr Din Varmepumpe vers. 2". Danish. Tech. report ForskEL 12075. Inero Energy, FEB 2015. <https://www.styrdinvarmepumpe.dk/sites/default/files/Slutrapportering%20fra%20SDVP2%20forskEL%2012075.pdf> (⌚ JUN 15, 2015). p. 271
- [172] JONGERDEN Marijn R, HÜLS J, HAVERKORT BRHM & REMKE AKI. "Assessing the cost of energy independence". In *EnergyCon*. Leuven, Belgium: IEEE PES, APR 2016. ISBN 978-1-4673-8464-3. DOI 10.1109/ENERGYCON.2016.7514139. p. 10
- [173] JOSELIN HERBERT G. M, INIYAN S & AMUTHA D. "A review of technical issues on the development of wind farms". In *Renewable and sustainable energy reviews* v32, 2014, pp. 619–641. ISSN 1364-0321. DOI 10.1016/j.rser.2014.01.055. p. 243
- [174] JOSKOW Paul L. "Lessons learned from electricity market liberalization". In *The energy journal*, 2008. Special Issue: The Future of Electricity, pp. 9–42. ISSN 0195-6574. DOI 10.5547/ISSN0195-6574-EJ-Vol29-NoSI2-3. <http://economics.mit.edu/files/2093> (⌚ JUN 4, 2015). p. 255
- [175] JOSKOW Paul L & TIROLE J. "Transmission rights and market power on electric power networks". In *RAND journal of economics* v31: 3, 2000, pp. 450–487. <http://economics.mit.edu/files/1136> (⌚ MAY 12, 2015). p. 12

- [176] JOUW ENERGIE MOMENT. *Handleiding energiecomputer*. Dutch. Enexis. <http://jouwenergiemoment.nl/jouw-wijk/download/type/document/id/20> (⊙ JUN 6, 2015).
p. 264
- [177] JOUW ENERGIE MOMENT. *Handleiding slimme wasmachine*. Dutch. Enexis & Indesit. <http://jouwenergiemoment.nl/jouw-wijk/download/type/document/id/21> (⊙ JUN 6, 2015).
p. 264
- [178] JUELGAARD Morten, TOTU LC, SHAFIEI SE et al. “Control structures for smart grid balancing”. In *ISGT Europe*. Lyngby, Denmark: IEEE PES, Oct 2013. DOI 10.1109/ISGTEurope.2013.6695321.
p. 76
- [179] KADUREK Petr, BLOM J, COBBEN JFG & KLING WL. “Theft detection and smart metering practices and expectations in the Netherlands”. In *ISGT Europe*. Oct 2010. DOI 10.1109/ISGTEUROPE.2010.5638852.
p. 259
- [180] KALDELLIS John K & KAPSALI M. “Shifting towards offshore wind energy—recent activity and future development”. In *Energy policy* 53, 2013, pp. 136–148. ISSN 0301-4215. DOI 10.1016/j.enpol.2012.10.032.
p. 243
- [181] KALL Peter & WALLACE SW. “Stochastic programming”. 2nd edition. Chichester: John Wiley & Sons, 1994. ISBN 978-0-471-95108-7.
p. 16
- [182] KEMPKER Pia, DIJK Nv, SCHEINHARDT W et al. “Optimization of charging strategies for electric vehicles in PowerMatcher-driven smart energy grids”. In *Proceedings of the 9th EAI international conference on performance evaluation methodologies and tools*. VALUETOOLS'15. Berlin, Germany: Institute for Computer Sciences, Social-Informatics & Telecommunications Engineering (ICST), JAN 2016, pp. 242–249. ISBN 978-1-63190-096-9. DOI 10.4108/eai.4-1-2016.151091.
p. 28
- [183] KEMPTON Willett, TOMIĆ J, LETENDRE S et al. “Vehicle-to-Grid power: battery, hybrid, and fuel cell vehicles as resources for distributed electric power in California”. Tech. report UCD-ITS-RR-01-03. UC Davis, JUN 2001. <http://escholarship.org/uc/item/0qp6s4mb.pdf> (⊙ JUN 6, 2015).
p. 270
- [184] KEMPTON Willett, UDO V, HUBER K et al. “A test of vehicle-to-grid (V2G) for energy storage and frequency regulation in the PJM system, results from an industry–university research partnership”. Tech. report. University of Delaware, Nov 2008. <http://www.udel.edu/V2G/resources/test-v2g-in-pjm-jan09.pdf> (⊙ JUN 4, 2015).
p. 248
- [185] KHODAEI Amin. “Provisional microgrids”. In *Smart grid, IEEE transactions on* 6:3, MAY 2015, pp. 1107–1115. ISSN 1949-3053. DOI 10.1109/TSG.2014.2358885.
p. 22
- [186] KHODR Hussein M, MARTINEZ-CRESPO J, MATOS MA & PEREIRA J. “Distribution systems reconfiguration based on OPF using Benders decomposition”. In *Power delivery, IEEE transactions on* 24:4, OCT 2009, pp. 2166–2176. ISSN 0885-8977. DOI 10.1109/TPWRD.2009.2027510.
p. 22
- [187] KIM Jin-Ho & SHCHERBAKOVA A. “Common failures of demand response”. In *Energy* 36:2, 2011, pp. 873–880. ISSN 0360-5442. DOI 10.1016/j.energy.2010.12.027.
p. 18
- [188] KIT. “Hybrid energy grid management”. EIT ICT Labs/Digital project. 2013–2015. <http://www.aifb.kit.edu/web/HEGRID/en> (⊙ SEP 18, 2014).
p. 227
- [189] KLAASSEN Elke, ZHANG Y, LAMPROPOULOS I & SLOOTWEG H. “Demand side management of electric boilers”. In *ISGT Europe*. Oct 2012. DOI 10.1109/ISGTEurope.2012.6465681.
pp. 18, 268
- [190] van der KLAUW Thijs, GERARDS MET, SMIT GJM & HURINK JL. “Optimal scheduling of electrical vehicle charging under two types of steering signals”. In *ISGT Europe*. Istanbul, Turkey: IEEE PES, Oct 2014. ISBN 978-1-4799-7720-8. DOI 10.1109/ISGTEurope.2014.7028746.
pp. 111, 309

- [191] van der KLAUW Thijs, GERARDS MET, SMIT GJM & HURINK JL. "Resource allocation problems in decentralized energy management". In *OR spectrum*, 2016. Submitted. ISSN 0171-6468. p. 305
- [192] KLIMA, ENERGI- OG BYGNINGSMINISTERIET. "Vindmøller slog rekord i 2014". *Danish*. JAN 2015. <http://www.kebmin.dk/nyheder/vindmoeller-slog-rekord-2014> (⌚ MAR 30, 2015). p. 240
- [193] KOCH Stephan, MATHIEU JL & CALLAWAY DS. "Modeling and control of aggregated heterogeneous thermostatically controlled loads for ancillary services". In *Proceedings of the 17th power systems computation conference*. Stockholm, Sweden, AUG 2011. http://www.psc-central.org/uploads/tx_ethpublications/fp133_01.pdf (⌚ JUN 7, 2015). p. 24
- [194] KOK J Koen AHPJ. "Short-term economics of virtual power plants". In *20th international conference and exhibition on electricity distribution (CIRED)*. Prague, Czech Republic, 2009. ISBN 978-1-84919-126-5. p. 28
- [195] KOK J Koen AHPJ. "The PowerMatcher: smart coordination for the smart electricity grid". PhD thesis. Vrije Universiteit Amsterdam, JUL 2013. <http://hdl.handle.net/1871/43567> (⌚ SEP 19, 2016). pp. 24, 27, 28, 30, 36, 188, 191, 260
- [196] KOKOPELI Peter, SCHREIFELS J & FORTE R. "Assessment of startup period at coal-fired electric generating units – revised". Tech. report. US Environmental Protection Agency, Office of Air & Radiation, Nov 2014. <http://www.epa.gov/ttnatw01/utility/matssfinalrulesd110414.pdf> (⌚ MAY 4, 2015). p. 9
- [197] KOOLWIJK Erik, GOUDSWAARD P, DIJKSHOORN A & OPREL L. "WKK in Het Nieuwe Telen". *Dutch*. Tech. report 09.399. Cogen, JAN 2010. p. 252
- [198] KRIST Jan Oene. "The effect of residential storage and control on the distribution net compared to central storage and control". MSc thesis. JUN 2014. <http://essay.utwente.nl/65437/> (⌚ MAY 27, 2015). pp. 34, 38, 71, 74
- [199] KU LEUVEN ENERGY INSTITUTE. "Negative electricity market prices". Tech. report 2014-01. ELECTA, 2014. p. 241
- [200] KWON Roy H & FRANCES D. "Optimization-based bidding in day-ahead electricity auction markets: a review of models for power producers". In *Handbook of networks in power systems I*. Edited by SOROKIN A, REBENNACK S, PARDALOS PM et al. Energy Systems. Springer, 2012, pp. 41–59. ISBN 978-3-642-23192-6. DOI 10.1007/978-3-642-23193-3_2. p. 15
- [201] LABADIE John W. "Optimal operation of multireservoir systems: state-of-the-art review". In *Journal of water resources planning and management* v130 : 2, 2004, pp. 93–111. DOI 10.1061/(ASCE)0733-9496(2004)130:2(93). p. 244
- [202] LARSEN Hans H & PETERSEN LS. "Wind energy – drivers and barriers for higher shares in the global power generation mix". Tech. report. DTU International Energy Report 2014. DTU National Laboratory for Sustainable Energy, Nov 2014. ISBN 978-87-550-3969-8. p. 241
- [203] LATTNER Chris & ADVE V. "LLVM: a compilation framework for lifelong program analysis & transformation". In *Proceedings of the 2004 international symposium on code generation and optimization (CGO'04)*. Palo Alto, California, MAR 2004, pp. 75–86. DOI 10.1109/CGO.2004.1281665. p. 195
- [204] LAURENT Jean-Charles, DESAULNIERS G, MALHAME RP & SOUMIS F. "A column generation method for optimal load management via control of electric water heaters". In *Power systems, IEEE transactions on* v10 : 3, AUG 1995, pp. 1389–1400. ISSN 0885-8950. DOI 10.1109/59.466513. p. 76
- [205] LAZOPOULOU Marilena. "Pricing schemes for automated demand response programmes: a quantitative analysis and comparison of pricing schemes with a smart grid control system". MSc thesis. Utrecht University, AUG 2014. pp. 273, 293–295, 306

- [206] LEEMPUT Niels, GETH F, CLAESSENS BJ et al. "A case study of coordinated electric vehicle charging for peak shaving on a low voltage grid". In *ISGT Europe*. IEEE PES, OCT 2012. ISBN 978-1-4673-2595-0. DOI 10.1109/ISGTEurope.2012.6465656. pp. 23, 24, 27
- [207] LI Tao & SHAHIDEHPOUR M. "Price-based unit commitment: a case of Lagrangian relaxation versus mixed integer programming". In *Power systems, IEEE transactions on* v20 : 4, Nov 2005, pp. 2015–2025. ISSN 0885-8950. DOI 10.1109/TPWRS.2005.857391. p. 14
- [208] LIANDER. "Stresstest succesvol: Lochem zonder stroom". *Dutch*. Press release. APR 2015. <https://www.liander.nl/nieuws/2015/04/07/stresstest-succesvol-lochem-zonder-stroom> (⌚ APR 7, 2015). p. 248
- [209] LIEFERS Bart J, CLAESSENS FN, PAUWELS EJ et al. "Market Garden: a simulation environment for research and user experience in smart grids". In *Proceedings of international conference on practical applications of agents and multi-agent systems*. Springer, 2014, pp. 351–354. DOI 10.1007/978-3-319-07551-8_36. p. 30
- [210] LINEAR CONSORTIUM. "Demand response for families". Tech. report. OCT 2014. http://www.linear-smartgrid.be/sites/default/files/boekje_linear_okt_2014_boekje_web.pdf (⌚ MAY 2, 2015). pp. 264, 271
- [211] LINNEMANN C., ECHTERNACHT D, BREUER C & MOSER A. "Modeling optimal redispatch for the European transmission grid". In *PowerTech*. Trondheim, JUN 2011. DOI 10.1109/PTC.2011.6019442. p. 11
- [212] LIU Cong, SHAHIDEHPOUR M & WU L. "Extended Benders decomposition for two-stage SCUC". In *Power systems, IEEE transactions on* v25 : 2, MAY 2010, pp. 1192–1194. ISSN 0885-8950. DOI 10.1109/TPWRS.2009.2038019. p. 14
- [213] LOKHOV Alexey. "Load-following with nuclear power plants". Newsletter article. 2011. <https://www.oecd-neo.org/nea-news/2011/29-2/nea-news-29-2-load-following-e.pdf> (⌚ JUN 8, 2015). p. 241
- [214] LÜBBECKE Marco E & DESROSIERS J. "Selected topics in column generation". In *Operations research* v53 : 6, 2005, pp. 1007–1023. DOI 10.1287/opre.1050.0234. pp. 21, 90, 91, 105
- [215] LUCAS Caroline. "Solar has been too successful – that's why tariffs were cut". The Guardian. News article. Nov 2011. <http://www.theguardian.com/environment/2011/nov/04/solar-successful-tariffs-cut> (⌚ MAR 31, 2015). p. 242
- [216] LUND Henrik & SALGI G. "The role of compressed air energy storage (CAES) in future sustainable energy systems". In *Energy conversion and management* v50 : 5, 2009, pp. 1172–1179. ISSN 0196-8904. DOI 10.1016/j.enconman.2009.01.032. p. 258
- [217] MACDOUGALL Pamela, KOK JKAHPJ, WARMER C & ROOSSIEN B. "Flexibility dynamics in clusters of residential demand response and distributed generation". In *22nd international conference and exhibition on electricity distribution (CIRED)*. Stockholm, Sweden, JUN 2013. p. 28
- [218] MACDOUGALL Pamela, ROOSSIEN B, WARMER C & KOK K. "Quantifying flexibility for smart grid services". In *Power and energy society general meeting (PES GM)*. IEEE PES, JUL 2013. DOI 10.1109/PESMG.2013.6672817. p. 28
- [219] MAHARJAN Sabita, ZHU Q, ZHANG Y et al. "Dependable demand response management in the smart grid: a Stackelberg game approach". In *Smart grid, IEEE transactions on* v4 : 1, MAR 2013, pp. 120–132. ISSN 1949-3053. DOI 10.1109/TSG.2012.2223766. p. 30
- [220] MALCOLM Scott A & ZENIOS SA. "Robust optimization for power systems capacity expansion under uncertainty". In *Journal of the operational research society* v45 : 9, Palgrave Macmillan Journals on behalf of the Operational Research Society, 1994, pp. 1040–1049. ISSN 0160-5682. DOI 10.1057/jors.1994.169. pp. 10, 16

- [221] MARINESCU Bogdan, COULONDRE JM, TSAMASPHYROU P & BOURMAUD JY. "Improving TSO's coordination for cross-border redispatch: a discussion of possible approaches in the European context". In *CIGRE/IEEE PES international symposium*. OCT 2005, pp. 378–385. DOI 10.1109/CIGRE.2005.1532766. p. 11
- [222] MARNAY Chris, VENKATARAMANAN G, STADLER M et al. "Optimal technology selection and operation of commercial-building microgrids". In *Power systems, IEEE transactions on* v23:3, AUG 2008, pp. 975–982. ISSN 0885-8950. DOI 10.1109/TPWRS.2008.922654. p. 22
- [223] MARTINOT Eric. "How is Denmark integrating and balancing renewable energy today?" Article. JAN 2015. <http://www.martinot.info/renewables2050/how-is-denmark-integrating-and-balancing-renewable-energy-today> (⊗ MAR 31, 2015). p. 241
- [224] MATEO Carlos, FRÍAS P, RESESES J et al. "PV Grid – European advisory paper". Tech. report. PV Grid project, JUL 2014. http://www.pvgrid.eu/fileadmin/PVgrid_AdvisPaper.pdf (⊗ SEP 19, 2016). p. 265
- [225] McNAMARA Paul & McLOONE S. "Hierarchical demand response for peak minimization using Dantzig–Wolfe decomposition". In *Smart grid, IEEE transactions on* v6:6, Nov 2015, pp. 2807–2815. ISSN 1949-3053. DOI 10.1109/TSG.2015.2467213. pp. 21, 76
- [226] McNAMARA Paul & McLOONE S. "Hierarchical demand response using Dantzig–Wolfe decomposition". In *ISGT Europe*. Lyngby, Denmark: IEEE PES, OCT 2013. DOI 10.1109/ISGTEurope.2013.6695269. p. 21, 76
- [227] MEINSHAUSEN Malte, MEINSHAUSEN N, HARE W et al. "Greenhouse-gas emission targets for limiting global warming to 2 °C". In *Nature* v458, APR 2009, pp. 1158–1162. DOI 10.1038/nature08017. <http://www.nature.com/nature/journal/v458/n7242/full/nature08017.html> (⊗ MAY 28, 2015). p. 239
- [228] MERLIN André & SANDRIN P. "A new method for unit commitment at Electricité De France". In *Power apparatus and systems, IEEE transactions on* vPAS-102:5, MAY 1983, pp. 1218–1225. ISSN 0018-9510. DOI 10.1109/TPAS.1983.318063. pp. 13, 14, 42, 244
- [229] MIELE. "‘Smart Home’: Miele consolidates and expands its leading position". Press release. SEP 2014. <http://www.miele.de/en/m/smart-home-miele-consolidates-and-expands-its-leading-position-2482.htm> (⊗ JUN 6, 2015). p. 264
- [230] MOHSENIAN-RAD Amir-Hamed, WONG VWS, JATSKEVICH J et al. "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid". In *Smart grid, IEEE transactions on* v1:3, DEC 2010, pp. 320–331. ISSN 1949-3053. DOI 10.1109/TSG.2010.2089069. p. 30
- [231] MOLDERINK Albert. "On the three-step control methodology for smart grids". PhD thesis. University of Twente, MAY 2011. ISBN 978-90-365-3170-2. DOI 10.3990/1.9789036531702. pp. 30, 32, 33, 35, 36, 60, 72, 119, 252, 265, 288, 306
- [232] MOLDERINK Albert, BAKKER V, HURINK JL & SMIT GJM. "Comparing demand side management approaches". In *ISGT Europe*. Berlin, Germany: IEEE PES, OCT 2012. ISBN 978-1-4673-2596-7. DOI 10.1109/ISGTEurope.2012.6465699. pp. 28, 35, 37, 134
- [233] MOLDERINK Albert, BAKKER V, BOSMAN MGC et al. "Management and control of domestic smart grid technology". In *Smart grid, IEEE transactions on* v1:2, IEEE PES, SEP 2010, pp. 109–119. ISSN 1949-3053. DOI 10.1109/TSG.2010.2055904. pp. 44, 66, 67
- [234] MOMOH James A, ADAPA R & EL-HAWARY ME. "A review of selected optimal power flow literature to 1993, part I: nonlinear and quadratic programming approaches". In *Power systems, IEEE transactions on* v14:1, FEB 1999, pp. 96–104. ISSN 0885-8950. DOI 10.1109/59.744492. pp. 11, 13, 14, 244

- [235] MOMOH James A, EL-HAWARY ME & ADAPA R. "A review of selected optimal power flow literature to 1993, part II: Newton, linear programming and interior point methods". In *Power systems, IEEE transactions on* ^v14:1, FEB 1999, pp. 105–111. ISSN 0885-8950. DOI 10.1109/59.744495. p. 14
- [236] MORROW Kevin, KARNER D & FRANCFORT J. "Plug-in hybrid electric vehicle charging infrastructure review". Tech. report INL/EXT-08-15058. U.S. Department of Energy Vehicle Technologies Program – Advanced Vehicle Testing Activity, Nov 2008. <https://avt.inl.gov/sites/default/files/pdf/phev/phevInfrastructureReport08.pdf> (⌚ MAY 26, 2016). p. 277
- [237] MUNARI Pedro. "Theoretical and computational issues for improving the performance of linear optimization methods". PhD thesis. Universidade de São Paulo, 2013. <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-16042013-101426/publico/tesePedroAugustoMunariJunior.pdf> (⌚ JUN 13, 2016). p. 106
- [238] MUNARI Pedro & GONDZIO J. "Using the primal-dual interior point algorithm within the branch-price-and-cut method". In *Computers & operations research* ^v40:8, 2013, pp. 2026–2036. ISSN 0305-0548. DOI 10.1016/j.cor.2013.02.028. p. 106
- [239] NEDAP. "Solar Energie intelligent nutzen". German. <http://www.powerrouter.com/> (⌚ MAR 19, 2015). p. 266
- [240] NEDERLANDSE MEDEDELINGS-AUTORITEIT. "Beleidsregel redelijke terugleververgoedingen vergunninghouders elektriciteit". Dutch. Staatscourant. Case 102252-1. APR 2006. <https://zoek.officielebekendmakingen.nl/stcrt-2006-82-p28-SC74915.html> (⌚ MAR 20, 2015). p. 242
- [241] NEN. "Minimum set of functions for metering of electricity, gas and thermal energy for domestic customers". Dutch. Standard. AUG 2007. <https://www.nen.nl/NEN-Shop-2/Standard/NTA-81302007-nl.htm> (⌚ JUN 5, 2015). p. 259
- [242] NEN-EN. "Voltage characteristics of electricity supplied by public distribution networks". Standard. AUG 2010. <https://www.nen.nl/NEN-Shop-2/Standard/NENEN-501602010-en.htm> (⌚ MAY 29, 2015). pp. 254, 257
- [243] NEW YORK INDEPENDENT SYSTEM OPERATOR. "Blackout August 14, 2003 final report". Tech. report. FEB 2005. http://www.nyiso.com/public/webdocs/media_room/press_releases/2005/blackout_rpt_final.pdf (⌚ JUN 6, 2015). p. 258
- [244] NOËL Marie-Cécile & SMEERS Y. "Nested decomposition of multistage nonlinear programs with recourse". In *Mathematical programming* ^v37:2, Springer, 1987, pp. 131–152. ISSN 0025-5610. DOI 10.1007/BF02591691. p. 82
- [245] NORD POOL SPOT. "Block order". Web site. <http://www.nordpoolspot.com/TAS/Day-ahead-market-Elspot/Order-types/Block-bid/> (⌚ JUN 8, 2015). p. 13
- [246] NUTMA Jorrit. "More comprehensive demand side management by the integration of the PowerMatcher and Triana". MSc thesis. University of Twente, MAY 2016. pp. 38, 188
- [247] NYKAMP Stefan. "Integrating renewables in distribution grids: storage, regulation and the interaction of different stakeholders in future grids". PhD thesis. Enschede: University of Twente, OCT 2013. ISBN 978-90-365-0057-9. DOI 10.3990/1.9789036500579. pp. 15, 243, 248, 253, 255, 256, 258, 265
- [248] NYKAMP Stefan, BAKKER V, MOLDERINK A et al. "Break-even analysis for the storage of PV in power distribution grids". In *International journal of energy research* ^v38:9, Wiley, 2014, pp. 1112–1128. ISSN 0363-907x. DOI 10.1002/er.3106. p. 257
- [249] NYKAMP Stefan, BOSMAN MGC, MOLDERINK A et al. "Value of storage in distribution grids – competition or cooperation of stakeholders?" In *Smart grid, IEEE transactions on* ^v4:3, IEEE, SEP 2013, pp. 1361–1370. ISSN 1949-3053. DOI 10.1109/TSG.2013.2254730. pp. 253, 258

- [250] NYKAMP Stefan, FLOSBACH P, KÜPPERS S et al. "Storage options for distribution system operators to cope with photovoltaic and wind generation". In *International renewable energy storage conference (IRES)*. MAR 2015. p. 258
- [251] NYKAMP Stefan, MOLDERINK A, HURINK JL & SMIT GJM. "Statistics for PV, wind and biomass generators and their impact on distribution grid planning". In *Energy* ^v45:1, Elsevier, SEP 2012, pp. 924–932. ISSN 0360-5442. DOI 10.1016/j.energy.2012.06.067. p. 243
- [252] OPENADR ALLIANCE. "OpenADR 2.0 profile specification, A profile". Document 20110712-1. AUG 2012. <http://www.openadr.org/specification> (⌚ SEP 19, 2016). p. 192
- [253] OPENADR ALLIANCE. "OpenADR 2.0 profile specification, B profile". Document 20120912-1. JAN 2013. <http://www.openadr.org/specification> (⌚ SEP 19, 2016). p. 193
- [254] OPENHAB. "Empowering the smart home". <http://www.openhab.org/> (⌚ MAR 12, 2015). p. 193
- [255] den OUDEN Bert, MOSER A & van der HOOFD E. "Market review 2014 H1". Tech. report. TenneT, SEP 2014. http://www.tennet.eu/fileadmin/user_upload/Company/Publications/Technical_Publications/Dutch/2014_H1_TenneT_Market_Review.pdf (⌚ AUG 28, 2016). pp. 10, 242
- [256] PALENSKY Peter & DIETRICH D. "Demand side management: demand response, intelligent energy systems, and smart loads". In *Industrial informatics, IEEE transactions on* ^v7:3, AUG 2011, pp. 381–388. ISSN 1551-3203. DOI 10.1109/TII.2011.2158841. p. 18
- [257] PALOMAR Daniel P & CHIANG M. "A tutorial on decomposition methods for network utility maximization". In *Selected areas in communications, IEEE journal on* ^v24:8, IEEE, 2006, pp. 1439–1451. ISSN 0733-8716. DOI 10.1109/JSAC.2006.879350. p. 20
- [258] PARIKH Neal & BOYD S. "Proximal algorithms". In *Foundations and trends in optimization* ^v1:3, Now Publishers Inc., JAN 2014, pp. 127–239. ISSN 2167-3888. DOI 10.1561/2400000003. https://web.stanford.edu/~boyd/papers/pdf/prox_algs.pdf (⌚ JUN 1, 2015). pp. 21, 106
- [259] PENNINGS Maarten C. "Een slimme koelkast voor slimme netten". *Dutch*. Magazine article, TVVL magazine. APR 2011. <http://www.fampennings.nl/maarten/boeken/tvvl.pdf> (⌚ AUG 28, 2016). p. 26
- [260] PÉREZ-ARRIAGA Ignacio J, OLMOS L & RIVIER M. "Financial transmission rights". Edited by ROSELLÓN J & KRISTIANSEN T. Vol. 7. Lecture Notes in Energy. Springer, 2013. Chapter Transmission pricing, pp. 49–76. ISBN 978-1-4471-4786-2. DOI 10.1007/978-1-4471-4787-9. pp. 26, 27
- [261] PERSSON Michael. "Elektriciteitscentrale Rotterdam is overbodig en wordt verscheept naar Azië". *Dutch*. News article. MAY 2013. <http://www.volkskrant.nl/economie/elektriciteitscentrale-rotterdam-is-overbodig-en-wordt-verscheept-naar-azie~a3436389/> (⌚ MAY 29, 2015). p. 241
- [262] PIPATTANASOMPORN Manisa, FEROZE H & RAHMAN S. "Multi-agent systems in a distributed smart grid: design and implementation". In *Power systems conference and exposition (PSCE)*. IEEE PES, MAR 2009. DOI 10.1109/PSCE.2009.4840087. p. 30
- [263] PJM. "Demand resource saturation analysis". Report. MAY 2010. <http://pjm.com/~media/committees-groups/committees/pc/20100811/20100811-item-10-demand-response-saturation-report.aspx> (⌚ JUN 8, 2015). p. 17
- [264] POWERMATCHERSUITE. "Downloads". MAY 2015. <http://flexiblepower.github.io/downloads/> (⌚ MAY 25, 2015). p. 198

- [265] PRINS Yvonne. “Dynamic programming in the PowerMatcher”. MSc thesis. University of Twente & TNO, JUN 2015. http://essay.utwente.nl/67145/1/Prins_MA_EEMCS.pdf (⌚ JUN 8, 2016). pp. 28, 135, 189, 235
- [266] van PRUISSSEN Olaf P & KAMPHUIS IG. “Grote concentraties warmtepompen in een woonwijk en gevolgen elektriciteitsnetwerk”. *Dutch*. Tech. report ECN-E-10-088. ECN, 2010. p. 247
- [267] van PRUISSSEN Olaf P & KAMPHUIS IG. “High concentration of heat pumps in suburban areas and reduction of their impact on the electricity network”. In *PowerTech*. Trondheim: IEEE PES, 2011. DOI 10.1109/PTC.2011.6019171. p. 247
- [268] van PRUISSSEN Olaf P, van der TOGT A & WERKMAN E. “Energy efficiency comparison of a centralized and a multi-agent market based heating system in a field test”. In *Energy procedia* 62, 2014, pp. 170–179. ISSN 1876-6102. DOI 10.1016/j.egypro.2014.12.378. p. 58
- [269] PUBLIC SERVICE COMMISSION OF WISCONSIN. “Common power plant siting criteria”. Overview. SEP 1999. <http://psc.wi.gov/thelibrary/publications/electric/electric05.pdf> (⌚ JUN 15, 2015). p. 248
- [270] QURRENT, VERENIGING EIGEN HUIS & LIANDER. “VEH en Qurrent starten proef met slimme metercontract”. *Dutch*. Nov 2015. <https://www.qurrent.nl/updates/bericht/1261/veh-en-qurrent-starten-proef-met-slimme-metercontract> (⌚ MAY 25, 2016). p. 197
- [271] RAGHAVAN Prabhakar & THOMPSON CD. “Randomized rounding: a technique for provably good algorithms and algorithmic proofs”. In *Combinatorica* 7: 4, 1987, pp. 365–374. ISSN 0209-9683. DOI 10.1007/BF02579324. p. 93
- [272] RAMCHURN Sarvapali D, VYTELINGUM P, ROGERS A & JENNINGS NR. “Putting the ‘smarts’ into the smart grid: a grand challenge for artificial intelligence”. In *Communication of the ACM* 55: 4, ACM, APR 2012, pp. 86–97. ISSN 0001-0782. DOI 10.1145/2133806.2133825. p. 29
- [273] REZAEI Behnaz & ROSEN MA. “District heating and cooling: review of technology and potential enhancements”. In *Applied energy* 93, 2012, pp. 2–10. ISSN 0306-2619. DOI 10.1016/j.apenergy.2011.04.020. p. 251
- [274] RICHARDSON Peter, FLYNN D & KEANE A. “Local versus centralized charging strategies for electric vehicles in low voltage distribution systems”. In *Smart grid, IEEE transactions on* 3: 2, JUN 2012, pp. 1020–1028. ISSN 1949-3053. DOI 10.1109/TSG.2012.2185523. p. 18
- [275] RUELENS Frederik, VANDAELE S, LETERME W et al. “Demand side management of electric vehicles with uncertainty on arrival and departure times”. In *ISGT Europe*. IEEE PES, OCT 2012. DOI 10.1109/ISGTEurope.2012.6465695. pp. 23, 24
- [276] RUSH Alexander M & COLLINS M. “A tutorial on dual decomposition and Lagrangian relaxation for inference in natural language processing”. In *Journal of artificial intelligence research* 45: 1405.5208, APR 2012, pp. 305–362. ISSN 1076-9757. DOI 10.1613/jair.3680. <http://arxiv.org/abs/1405.5208> (⌚ MAY 30, 2015). p. 20
- [277] RUSSELL Stuart J & NORVIG P. “Artificial intelligence: a modern approach”. 2nd edition. Pearson Education, 2003. ISBN 978-0-13-790395-5. pp. 28, 29
- [278] RWE. “Der Smart Operator in der Pilotkommune Schwabmünchen”. *German*. Web site. <http://www.rwe.com/web/cms/de/2467056/rwe-deutschland-ag/energiawende/intelligente-netze/smart-operator/modellregionen/schwabmuenchen/> (⌚ JUN 15, 2015). p. 271
- [279] RWE. “In der Kita Haren denkt die Waschmaschine mit”. *German*. Web site. <http://www.rwe.com/web/cms/de/2257946/rwe-deutschland-ag/energiawende/intelligente-netze/smart-stations/pilotprojekt-kita-haren/> (⌚ JUN 15, 2015). p. 271

- [280] RWE. "RWE posts first net loss in 60 years". Press release. MAR 2014. <http://www.rwe.com/web/cms/en/113648/rwe/press-news/press-release/?pmid=4010679> (⌚ MAR 31, 2015).
p. 241
- [281] RWE. "RWE-Mobility – driving into the future with electric cars". Web site. <http://www.rwe.com/web/cms/en/183210/rwe/innovation/projects-technologies/energy-application/e-mobility/> (⌚ JUN 4, 2015).
p. 248
- [282] SAMADI Pedram, MOHSENIAN-RAD AH, SCHOBBER R et al. "Optimal real-time pricing algorithm based on utility maximization for smart grid". In *Smart grid communications (SmartGrid-Comm), 2010 first IEEE international conference on*. OCT 2010, pp. 415–420. DOI 10.1109/SMARTGRID.2010.5622077.
pp. 21, 30
- [283] SARAVANAN B., DAS S, SIKRI S & KOTHARI DP. "A solution to the unit commitment problem – a review". In *Frontiers in energy* 7 : 2, SP Higher Education Press, 2013, pp. 223–236. ISSN 2095-1701. DOI 10.1007/s11708-013-0240-3.
p. 11
- [284] SCATTOLINI Riccardo. "Architectures for distributed and hierarchical model predictive control – a review". In *Journal of process control* 19 : 5, 2009, pp. 723–731. ISSN 0959-1524. DOI 10.1016/j.jprocont.2009.02.003.
p. 76
- [285] SCHEPERS Benno L & van VALKENGOED MPJ. "Warmtenetten in Nederland, overzicht van grootschalige en kleinschalige warmtenetten in Nederland". Dutch. Tech. report 09.3031.45. CE Delft, OCT 2009.
p. 251
- [286] SCHNEIDER Kevin P, FULLER JC & CHASSIN D. "Analysis of distribution level residential demand response". In *Power systems conference and exposition (PSCE)*. IEEE PES, MAR 2011. DOI 10.1109/PSCE.2011.5772572.
pp. 17, 18
- [287] SEEBREGTS Ad. "Effecten van versneld sluiten van de vijf oudste kolencentrales". Dutch. Tech. report ECN-N-13-034. ECN, SEP 2013. <https://www.acm.nl/nl/download/bijlage/?id=11544> (⌚ MAY 29, 2015).
p. 241
- [288] SERNÉ Ronald. "Nulwoning". Dutch. <http://www.nulwoning.nl/> (⌚ JUL 11, 2016).
p. 245
- [289] SHAHIDEHOPOUR Mohammad & FU Y. "Benders decomposition: applying Benders decomposition to power systems". In *Power and energy magazine, IEEE* 3 : 2, MAR 2005, pp. 20–21. ISSN 1540-7977. DOI 10.1109/MPAE.2005.1405865.
p. 22
- [290] SHAHIDEHOPOUR Mohammad & FU Y. "Benders decomposition in restructured power systems". Tutorial. MAR 2005. <http://motor.ece.iit.edu/ms/benders.pdf> (⌚ JUN 8, 2015).
p. 22
- [291] SHAHIDEHOPOUR Mohammad & MARWALI M. "Maintenance scheduling in restructured power systems". Springer, 2000. ISBN 978-1-4613-7015-4. DOI 10.1007/978-1-4615-4473-9. <https://books.google.nl/books?hl=en&lr=&id=LyzoBwAAQBAJ> (⌚ JUN 8, 2015).
p. 10
- [292] SHAO Chengcheng, WANG X, WANG X et al. "Hierarchical charge control of large populations of EVs". In *Smart grid, IEEE transactions on* 7 : 2, MAR 2016, pp. 1147–1155. ISSN 1949-3053. DOI 10.1109/TSG.2015.2396952.
p. 22
- [293] SHERALI Hanif D & SMITH JC. "Improving discrete model representations via symmetry considerations". In *Management science* 47 : 10, INFORMS, OCT 2001, pp. 1396–1407. ISSN 0025-1909. DOI 10.1287/mnsc.47.10.1396.10265.
p. 14
- [294] SHRESTHA Govinda B, SONG K & GOEL L. "Strategic self-dispatch considering ramping costs in deregulated power markets". In *Power systems, IEEE transactions on* 19 : 3, AUG 2004, pp. 1575–1581. ISSN 0885-8950. DOI 10.1109/TPWRS.2004.825891.
p. 12
- [295] SIANO Pierluigi. "Demand response and smart grids – a survey". In *Renewable and sustainable energy reviews* 30, 2014, pp. 461–478. ISSN 1364-0321. DOI 10.1016/j.rser.2013.10.022.
pp. 18, 22

- [296] SIDDIQUI Sauleh, AZARM S & GABRIEL S. "A modified Benders decomposition method for efficient robust optimization under interval uncertainty". In *Structural and multidisciplinary optimization* v44 : 2, Springer, 2011, pp. 259–275. ISSN 1615-147x. DOI 10.1007/s00158-011-0631-1. p. 22
- [297] SILBERSCHATZ Abraham, GALVIN PB & GAGNE G. "Operating system concepts". 7th edition. John Wiley & Sons, 2005. ISBN 978-0-471-69466-3. p. 117
- [298] SIMONS Wessel. "Saldering na 2017 hoogst onzeker". *Dutch. Energiebusiness*. Nov 2013. <http://www.energiebusiness.nl/2013/11/12/saldering-na-2017-hoogst-onzeker/> (⌚ MAR 31, 2015). p. 242
- [299] SIOSHANSI Ramteen, OREN S & O'NEILL R. "The cost of anarchy in self-commitment based electricity markets". In *Competitive electricity markets: design, implementation and performance*. Edited by SIOSHANSI FP. 1st edition. Elsevier, 2008. Chapter 6, pp. 245–266. ISBN 978-0-08-047172-3. <http://www.ieor.berkeley.edu/~oren/pubs/III.B.2.pdf> (⌚ MAY 12, 2015). p. 15
- [300] SLYKE Richard MV & WETS R. "L-shaped linear programs with applications to optimal control and stochastic programming". In *SIAM journal on applied mathematics* v17 : 4, 1969, pp. 638–663. ISSN 0025-5610. DOI 10.1137/0117061. pp. 16, 22
- [301] SOCIAAL-ECONOMISCHE RAAD. "Energieakkoord – informatie voor belanghebbenden". *Dutch*. White paper. <http://www.energieakkoordser.nl/~media/files/energieakkoord/overzicht-belangrijkste-maatregelen-energieakkoord.aspx> (⌚ MAY 29, 2015). pp. 240, 241
- [302] SÖDER Lennart, ABILDGAARD H, ESTANQUEIRO A et al. "Experience and challenges with short-term balancing in European systems with large share of wind power". In *Sustainable energy, IEEE transactions on* v3 : 4, Oct 2012, pp. 853–861. ISSN 1949-3029. DOI 10.1109/TSTE.2012.2208483. p. 241
- [303] SOKOLER Leo E, EDLUND K, STANDARDI L & JØRGENSEN JB. "A decomposition algorithm for optimal control of distributed energy system". In *ISGT Europe*. Lyngby, Denmark: IEEE PES, Oct 2013. pp. 21, 76
- [304] SOLIMAN Soliman A & MANTAWY AH. "Economic Dispatch (ED) and Unit Commitment Problems (UCP): formulation and solution algorithms". In *Modern optimization techniques with applications in electric power systems*. Energy Systems. Springer, 2012, pp. 185–279. ISBN 978-1-4614-1751-4. DOI 10.1007/978-1-4614-1752-1. p. 11
- [305] SOROUDI Alireza. "Robust optimization based self scheduling of hydro-thermal Genco in smart grids". In *Energy* v61, 2013, pp. 262–271. ISSN 0360-5442. DOI 10.1016/j.energy.2013.09.014. p. 16
- [306] STANDARDI Laura, POULSEN NK, JØRGENSEN JB & SOKOLER LE. "Computational efficiency of economic MPC for power systems operation". In *ISGT Europe*. Lyngby, Denmark: IEEE PES, Oct 2013. pp. 21, 76
- [307] STEDIN. "Elektriciteits tarieven 2013, aansluiting en transport voor grootverbruikers". *Dutch*. INF.TAE.GVB.001.01.13. 2013. <https://www.stedin.net/zakelijk/~media/files/stedin/tarieven/gv/stedin-2013-tarieven-gv-elektriciteit.pdf> (⌚ MAY 28, 2016). p. 294
- [308] STEDIN. "Europese subsidie voor innovatieprogramma nul-op-de-meter-woningen". Press release. MAY 2015. <https://www.stedin.net/over-stedin/pers-en-media/persberichten/europese-subsidie-voor-innovatieprogramma-nul-op-de-meter-woningen> (⌚ JUN 10, 2015). p. 242
- [309] van STRATEN Gerrit, van WILLIGENBURG G, van HENTEN E & van OOTEGHEM R. "Optimal control of greenhouse cultivation". CRC Press, Taylor & Francis Group, 2011. ISBN 978-1-4200-5963-2. <https://books.google.nl/books?id=5UP01xxmXLIC> (⌚ JUN 8, 2015). p. 252

- [310] STREIFFERT Dan, PHILBRICK R & OTT A. "A mixed integer programming solution for market clearing and reliability analysis". In *Power engineering society general meeting (PES GM)*. Vol. 3. IEEE PES, JUN 2005, pp. 2724–2731. DOI 10.1109/PES.2005.1489108. pp. 13, 14, 244
- [311] STROOMVERSNELLING. "De Stroomversnelling, samenwerkingsovereenkomst". *Dutch*. Agreement. Version 10.4. JUN 2013. <http://energiesprong.nl/wp-content/uploads/2013/06/Stroomversnelling-Samenwerkingsovereenkomst.pdf> (⌚ JUN 10, 2015). p. 242
- [312] STROOMVERSNELLING. "Stroomversnelling Nederland: 4,5 miljoen woningen naar nul-op-de-meter, hoe gemeenten eenvoudig kunnen bijdragen". *Dutch*. Brochure. FEB 2015. <http://www.energielinq.nl/uploads/attachment/file/6/36/Brochure-Taskforce-Vergunningen-1427815887.pdf> (⌚ JUN 10, 2015). p. 242
- [313] STRUTH Janina, KAIRIES K, LEUTHOLD M et al. "PV-benefit: a critical review of the effect of grid integrated PV-storage-systems". In *International renewable energy storage conference (IRES)*. Nov 2013. http://www.pv-nutzen.rwth-aachen.de/wp-content/uploads/2013/05/Struth_IRES13.pdf (⌚ MAY 11, 2015). pp. 266, 267
- [314] SU Wencong, EICHI H, ZENG W & CHOW M. "A survey on the electrification of transportation in a smart grid environment". In *Industrial informatics, IEEE transactions on* '8: 1, FEB 2012, pp. 1–10. ISSN 1551-3203. DOI 10.1109/TII.2011.2172454. p. 248
- [315] TENNET. "BRP register". <http://www.tennet.eu/electricity-market/dutch-market/brp-register/> (⌚ SEP 19, 2016). p. 12
- [316] TENNET. "Samenvatting, conclusies en aanbevelingen storing Diemen". *Dutch*. Tech. report. JUN 2015. http://www.tennet.eu/fileadmin/user_upload/Company/News/English/2015/20150611_Samenvatting_rapporten_storing_Diemen_DEF.pdf (⌚ AUG 28, 2016). p. 254
- [317] TESLA MOTORS. "Powerwall: energy storage for a sustainable home". APR 2015. <http://www.teslamotors.com/powerwall> (⌚ MAY 10, 2015). p. 266
- [318] THIELE Aurélie, TERRY T & EPELMAN M. "Robust linear optimization with recourse". Tech. report TR09-01. University of Michigan, Ann Arbor, MAR 2010. http://www.optimization-online.org/DB_FILE/2009/03/2263.pdf (⌚ JUN 16, 2015). p. 22
- [319] THOLEN Michael & WALKER-HERTKORN S. "Geothermie: Grundlagen für oberflächennahe Erdwärmesondenbohrungen". *German*. Wirtschafts- und Verlagsgesellschaft Gas und Wasser, 2008. ISBN 978-3-8955-4167-4. p. 246
- [320] TNO, ALLIANDER, iNRG & TECHNOLUTION. "Topsector energie: virtual infrastructure operating system (VIOS)". *Dutch*. TKIS2SG2012.4. 2012–2015. <http://topsectorenergie.nl/projecten-post/virtual-infrastructure-operating-system-vios/> (⌚ MAY 25, 2016). p. 227
- [321] Tol Richard SJ. "A cost–benefit analysis of the EU 20/20/2020 package". In *Energy policy* '49, OCT 2012, pp. 288–295. ISSN 0301-4215. DOI 10.1016/j.enpol.2012.06.018. p. 240
- [322] TOPALIS Frangiskos, IRREK W & TARGOSZ R. "Strategies for development and diffusion of energy efficient distribution transformers". Project report. JUL 2008. https://ec.europa.eu/energy/intelligent/projects/sites/iee-projects/files/projects/documents/seedt_publishable_report_en.pdf (⌚ MAY 26, 2016). p. 282
- [323] TOYOTA. "Prius plug-in hybrid". 2015. <http://www.toyota.com/prius-plug-in-hybrid/> (⌚ APR 7, 2015). p. 248
- [324] TROESTER Eckehard. "New German grid codes for connecting PV systems to the medium voltage power grid". In *2nd international conference on concentrating photovoltaic power plants: optical design, production, grid connection*. MAR 2009. http://www.energynautics.com/downloads/publikationen/Paper_Energynautics_Connecting_PV_Systems.pdf (⌚ MAY 16, 2015). p. 18

- [325] TROUW. “GDF Suez sluit deel gascentrales Nederland”. *Dutch*. News article. JAN 2013. <http://www.trouw.nl/tr/nl/4332/Groen/article/detail/3377258/2013/01/15/GDF-Suez-sluit-deel-gascentrales-Nederland.dhtml> (⌚ MAY 28, 2015). p. 241
- [326] TSAO Jeffrey Y & WAIDE P. “The world’s appetite for light: empirical data and trends spanning three centuries and six continents”. In *Leukos* ^v6: 4, 2010, pp. 259–281. DOI 10.1582/LEUKOS.2010.06.04001. http://www.sandia.gov/~jytsao/tsao_jy_2010_04_app_for_light_LEUKOS.pdf (⌚ JUN 4, 2015). p. 245
- [327] UK DEPARTMENT OF ENERGY AND CLIMATE CHANGE. “Domestic energy consumption in the UK between 1970 and 2013”. In *Energy consumption in the UK (2014)*. Original link no longer available, mirrored at (http://web.archive.org/web/*/http://wwwhome.cs.utwente.nl/~toerscheha/phd/mirror/EnergyConsumptionUK2014.pdf, latest version at <https://www.gov.uk/government/statistics/energy-consumption-in-the-uk>. JUL 2014. Chapter 3. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/338662/ecuk_chapter_3_domestic_factsheet.pdf (⌚ JUN 4, 2015). p. 245
- [328] UK DEPARTMENT OF ENERGY & CLIMATE CHANGE. “Electricity market reform: capacity market”. JUL 2014. <https://www.gov.uk/government/collections/electricity-market-reform-capacity-market> (⌚ MAR 31, 2015). p. 241
- [329] UNITED NATIONS FRAMEWORK CONVENTION ON CLIMATE CHANGE. “Kyoto protocol to the United Nations framework convention on climate change”. 1998. <http://unfccc.int/resource/docs/convkp/kpeng.pdf> (⌚ MAY 28, 2015). p. 239
- [330] US ENERGY INFORMATION ADMINISTRATION. “Average retail price of electricity to ultimate customers by end-use sector”. http://www.eia.gov/electricity/monthly/epm_table_grapher.cfm?t=epmt_5_6_a (⌚ MAR 20, 2015). p. 240
- [331] VANDAELE Stijn, CLAESSENS BJ, HOMMELBERG MPF et al. “A scalable three-step approach for demand side management of plug-in hybrid vehicles”. In *Smart grid, IEEE Transactions on* ^v4: 2, JUN 2013, pp. 720–728. ISSN 1949-3053. DOI 10.1109/TSG.2012.2213847. pp. 23, 296
- [332] VANDERBECK François. “Decomposition and column generation for integer programs”. PhD thesis. Université Catholique de Louvain, SEP 1994. <https://www.math.u-bordeaux.fr/~fvanderb/papers/phd.pdf> (⌚ MAY 24, 2016). p. 78, 91
- [333] VANDERBECK François & WOLSEY LA. “An exact algorithm for IP column generation”. In *Operations research letters* ^v19: 4, 1996, pp. 151–159. ISSN 0167-6377. DOI 10.1016/0167-6377(96)00033-8. p. 78
- [334] VANTHOURNOUT Koen. “Demand response adaptor for connecting a device to a demand response system or home automation system, device and such demand response adaptor and use thereof”. US 20130184881 A1. JUL 2013. <https://www.google.com/patents/US20130184881>. p. 264
- [335] VARDAKAS John S, ZORBA N & VERIKOUKIS CV. “A survey on demand response programs in smart grids: pricing methods and optimization algorithms”. In *Communications surveys tutorials, IEEE* ^v17: 1, Q1 2015, pp. 152–178. ISSN 1553-877x. DOI 10.1109/COMST.2014.2341586. p. 19
- [336] van der VELDEN Nico JA & SMIT PX. “Energemonitor van de Nederlandse glastuinbouw 2013”. *Dutch*. Tech. report LEI 2014-025. LEI Wageningen UR, Nov 2014. ISBN 978-90-8615-690-0. p. 252

- [337] VELDMAN Else. “Power play: impacts of flexibility in future residential electricity demand on distribution network utilisation”. PhD thesis. Technische Universiteit Eindhoven, DEC 2013. ISBN 978-90-386-3513-2. DOI 10.6100/IR761384. <http://repository.tue.nl/761384> (⌚ MAY 29, 2015). p. 244
- [338] VENTOSA Mariano, BAÍLLO Á, RAMOS A & RIVIER M. “Electricity market modeling trends”. In *Energy policy* √33 : 7, 2005, pp. 897–913. ISSN 0301-4215. DOI 10.1016/j.enpol.2003.10.013. p. 15
- [339] VERBEECK Griet. “Optimisation of extremely low energy residential buildings”. PhD thesis. KU Leuven, MAY 2007. ISBN 978-90-5682-801-1. <https://lirias.kuleuven.be/handle/1979/852> (⌚ MAY 26, 2016). p. 278
- [340] VERMESAN Ovidiu, BLYSTAD L, ZAFALON R et al. “Internet of energy — connecting energy anywhere anytime”. In *Advanced microsystems for automotive applications*. Edited by MEYER G & VALLDORF J. VDI-Buch. Springer, 2011, pp. 33–48. ISBN 978-3-642-21380-9. DOI 10.1007/978-3-642-21381-6. p. 260
- [341] VINKE Maarten. “An approximate dynamic programming approach to the micro-CHP scheduling problem”. MSc thesis. University of Twente, AUG 2012. <http://essay.utwente.nl/62262/> (⌚ MAY 27, 2015). p. 33
- [342] van VOORST Peter & SIEKMAN P. “Analyse invloed warmtepompen op het elektriciteits-net”. *Dutch*. Presentation. Liander, OCT 2010. <http://deteuge2.nl/presentaties/allliander20101027/> (⌚ MAY 8, 2015). p. 247
- [343] de VRIES Sven & VOHRA RV. “Combinatorial auctions: a survey”. In *INFORMS journal on computing* √15 : 3, INFORMS, 2003, pp. 284–309. ISSN ISSN: 1091-9856. DOI 10.1287/ijoc.15.3.284.16077. p. 26
- [344] WALDSCHMIDT Wolfgang. “ABC der Wärmepumpe”. *German*. VWEW Energieverlag, 2007. ISBN 978-3-8022-0784-6. p. 246
- [345] WALRAS Leon. “Elements of pure economics, or the theory of social wealth”. Translated by JAFFÉ W. Routledge, 1954. DOI 10.4324/9781315888958. <http://www.gbv.de/dms/zbw/657540056.pdf> (⌚ JUN 3, 2015). p. 24
- [346] WANG Jonathan, BIVIJ I M & WANG WM. “Case studies of smart grid demand response programs in North America”. In *ISGT NA. IEEE PES*, JAN 2011. DOI 10.1109/ISGT.2011.5759162. pp. 18, 262
- [347] WANG Wenye & LU Z. “Cyber security in the smart grid: survey and challenges”. In *Computer networks* √57 : 5, 2013, pp. 1344–1371. ISSN 1389-1286. DOI 10.1016/j.comnet.2012.12.017. p. 10
- [348] WANG Yunpeng, SAAD W, HAN Z et al. “A game-theoretic approach to energy trading in the smart grid”. In *Smart grid, IEEE transactions on* √5 : 3, MAY 2014, pp. 1439–1450. ISSN 1949-3053. DOI 10.1109/TSG.2013.2284664. p. 30
- [349] WATT David A & BROWN DF. “Programming language processors in Java: compilers and interpreters”. Prentice Hall, 2000. ISBN 978-0-13025-786-4. p. 194
- [350] WECKX Sam, D’HULST R, CLAESSENS B & DRIESSENSAM J. “Multiagent charging of electric vehicles respecting distribution transformer loading and voltage limits”. In *Smart grid, IEEE transactions on* √5 : 6, Nov 2014, pp. 2857–2867. ISSN 1949-3053. DOI 10.1109/TSG.2014.2345886. pp. 24, 30
- [351] WEIDLICH Anke & VEIT D. “A critical survey of agent-based wholesale electricity market models”. In *Energy economics* √30 : 4, JUL 2008, pp. 1728–1759. ISSN 0140-9883. DOI 10.1016/j.eneco.2008.01.003. pp. 15, 30

- [352] WELLMAN Michael P. "A market-oriented programming environment and its application to distributed multicommodity flow problems". In *Journal of artificial intelligence research* v1, 1993, pp. 1–23. ISSN 1076-9757. DOI 10.1613/jair.2. p. 26
- [353] WHEELER David A. "SLOCCount". <http://www.dwheeler.com/sloccount/> (☞ MAY 25, 2016). p. 192
- [354] WIDERGREN Steve, FULLER J, MARINOVICI C & SOMANI A. "Residential transactive control demonstration". In *ISGT NA*. FEB 2014. DOI 10.1109/ISGT.2014.6816405. p. 27
- [355] WIJBENGA Jan Pieter, MACDOUGALL P, KAMPHUIS IG et al. "Multi-goal optimization in PowerMatching city: a smart living lab". In *ISGT Europe*. IEEE PES, OCT 2014. DOI 10.1109/ISGTEurope.2014.7028799. pp. 27, 28
- [356] WIKIPEDIA. "Air conditioning". http://en.wikipedia.org/wiki/Air_conditioning (☞ APR 7, 2015). p. 247
- [357] WIKIPEDIA. "Concentrated solar power". http://en.wikipedia.org/wiki/Concentrated_solar_power (☞ JUN 9, 2015). p. 243
- [358] WIKIPEDIA. "Cross-correlation". <https://en.wikipedia.org/wiki/Cross-correlation> (☞ MAR 10, 2016). p. 208
- [359] WIKIPEDIA. "Energy density". http://en.wikipedia.org/wiki/Energy_density (☞ APR 1, 2015). pp. 244, 247
- [360] WIKIPEDIA. "Feed-in tariffs in Germany". http://en.wikipedia.org/wiki/Feed-in_tariffs_in_Germany (☞ MAR 20, 2015). p. 240
- [361] WIKIPEDIA. "Plug-in electric vehicles in Norway". http://en.wikipedia.org/wiki/Plug-in_electric_vehicles_in_Norway (☞ APR 8, 2015). p. 248
- [362] WILLING Stefan, NILGES J, NYKAMP S et al. "Improving quality of supply and usage of assets in distribution grids by introducing a "smart operator"". In *22nd international conference and exhibition on electricity distribution (CIRED)*. Stockholm, Sweden, JUN 2013. ISBN 978-1-84919-732-8. DOI 10.1049/cp.2013.0892. pp. 30, 191, 200, 271
- [363] WINDING-SØRENSEN Jon & MERG PM. "Norges mest solgte bil". *Norwegian*. News article. BilNorge, SEP 2013. <http://www.bilnorge.no/artikkel.php?aid=41259> (☞ APR 8, 2015). p. 248
- [364] WIRTH Harry. "Recent facts about photovoltaics in Germany". Tech. report. Fraunhofer institute for solar energy systems (ISE), JAN 2015. <http://www.ise.fraunhofer.de/en/publications/veroeffentlichungen-pdf-dateien-en/studien-und-konzeptpapiere/recent-facts-about-photovoltaics-in-germany.pdf> (☞ MAY 15, 2015). pp. 242, 243
- [365] WOLAK Frank A. "Deregulation and interdependence in the Asia-Pacific region". Edited by ITO T & KRUEGER AO. Vol. 8. University of Chicago Press, JAN 2000. Chapter Market design and price behavior in restructured electricity markets: an international comparison, pp. 79–137. ISBN 978-0-226-38674-4. <http://www.nber.org/chapters/c8478> (☞ MAY 12, 2015). p. 12
- [366] WOLFRUM Philipp, KAUTZ M & SCHÄFER J. "Optimal control of combined heat and power units under varying thermal loads". In *Control engineering practice* v30, SEP 2014, pp. 105–111. ISSN 0967-0661. DOI 10.1016/j.conengprac.2013.08.014. pp. 22, 193, 269
- [367] WOOLDRIDGE Michael. "An introduction to multi-agent systems". Chichester: John Wiley & Sons, 2002. ISBN 978-0-471-49691-5. pp. 28, 29
- [368] WU Lei, SHAHIDEHPOUR M & LI T. "Stochastic security-constrained unit commitment". In *Power systems, IEEE transactions on* v22:2, MAY 2007, pp. 800–811. ISSN 0885-8950. DOI 10.1109/TPWRS.2007.894843. p. 16

- [369] YGGE Fredrik. "Market-oriented programming and its application to power load management". PhD thesis. Lund University, JUN 1998. ISBN 978-91-628-3055-7. p. 26
- [370] YGGE Fredrik, AKKERMANS H, ANDERSSON A et al. "The homebots system and field tests: a multi-commodity market for predictive load management". In *Proceedings of the fourth international conference and exhibition on the practical application of intelligent agents and multi-agents (PAAM'99)*. 1999, pp. 363–382. <http://user.it.uu.se/~arnea/ps/bot.pdf> (⌚ JUN 9, 2015). p. 30
- [371] YORK Dan & KUSHLER M. "Exploring the relationship between demand response and energy efficiency: a review of experience and discussion of key issues". Tech. report U052. American Council for an Energy-Efficient Economy, MAR 2005. <http://aceee.org/research-report/u052> (⌚ APR 22, 2015). p. 262
- [372] YORK Dan, KUSHLER M & WITTE P. "Examining the peak demand impacts of energy efficiency: a review of program experience and industry practices". Tech. report U072. American Council for an Energy-Efficient Economy, FEB 2007. <http://aceee.org/research-report/u072> (⌚ APR 7, 2015). pp. 245, 247, 262
- [373] ZAAL Vincent. "Network analysis of EV charging infrastructure in Amsterdam". MSc thesis. University of Utrecht, AUG 2011. <http://dspace.library.uu.nl/handle/1874/210842> (⌚ JUN 4, 2015). p. 248
- [374] ZAKARIAZADEHA Alireza, JADIDA S & SIANO P. "Multi-objective scheduling of electric vehicles in smart distribution system". In *Energy conversion and management* v79, MAR 2014, pp. 43–53. DOI 10.1016/j.enconman.2013.11.042. p. 22
- [375] ZENG Bo & ZHAO L. "Solving two-stage robust optimization problems using a column-and-constraint generation method". In *Operations research letters* v41 : 5, 2013, pp. 457–461. ISSN 0167-6377. DOI 10.1016/j.orl.2013.05.003. pp. 16, 22
- [376] ZHANG Huili L, BAEYENS J, DEGRÈVE J & CACÈRES G. "Concentrated solar power plants: review and design methodology". In *Renewable and sustainable energy reviews* v22, 2013, pp. 466–481. ISSN 1364-0321. DOI 10.1016/j.rser.2013.01.032. p. 243
- [377] ZHANG Zhenghao, GONG S, DIMITROVSKI AD & LI H. "Time synchronization attack in smart grid: impact and analysis". In *Smart grid, IEEE transactions on* v4 : 1, MAR 2013, pp. 87–98. ISSN 1949-3053. DOI 10.1109/TSG.2012.2227342. p. 317
- [378] ZHAO Xing, LUH PB & WANG J. "Surrogate gradient algorithm for Lagrangian relaxation". In *Journal of optimization theory and applications* v100 : 3, Kluwer Academic Publishers-Plenum Publishers, 1999, pp. 699–712. ISSN 0022-3239. DOI 10.1023/A:1022646725208. p. 21
- [379] ZIGBEE ALLIANCE. "Home automation public application profile 1.2". Document 05-3520-29. JUN 2013. <http://www.zigbee.org/zigbee-for-developers/applicationstandards/zigbeehomeautomation/> (⌚ SEP 19, 2016). p. 193
- [380] ZIGBEE ALLIANCE. "Smart Energy Profile 2 application protocol standard". Document 13-0200-00. APR 2013. <http://www.zigbee.org/zigbee-for-developers/applicationstandards/zigbeesmartenergy/> (⌚ SEP 19, 2016). p. 193
- [381] ZOU Xiaoyan. "Double-sided auction mechanism design in electricity based on maximizing social welfare". In *Energy policy* v37 : 11, 2009, pp. 4231–4239. ISSN 0301-4215. DOI 10.1016/j.enpol.2009.05.019. p. 24
- [382] Z-WAVE ALLIANCE. "Energy management". Web site. <http://z-wavealliance.org/energy-management/> (⌚ MAR 12, 2015). p. 193

PUBLICATIONS

- [^{HAT}:1] TOERSCHE Hermen A, BAKKER V, MOLDERINK A, NYKAMP S, HURINK JL & SMIT GJM. “Controlling the heating mode of heat pumps with the TRIANA three step methodology”. In *ISGT NA*. Washington DC, USA: IEEE PES, JAN 2012. ISBN 978-1-4577-2158-8. DOI 10.1109/ISGT.2012.6175662. p. 35

TRIANA can manage large groups of heat pumps.

- [^{HAT}:2] TOERSCHE Hermen A, NYKAMP S, MOLDERINK A, HURINK JL & SMIT GJM. “Controlling smart grid adaptivity”. In *ISGT Europe*. Berlin, Germany: IEEE PES, OCT 2012. ISBN 978-1-4673-2595-0. DOI 10.1109/ISGTEurope.2012.6465748. pp. 233, 305

Smart grid control strategies have to consider the structure of disturbances to respond appropriately. We demonstrate this in a wind feed-in scenario with central battery storage.

- [^{HAT}:3] TOERSCHE Hermen A, MOLDERINK A, HURINK JL & SMIT GJM. “Column generation based planning in smart grids using TRIANA”. In *ISGT Europe*. Lyngby, Denmark: IEEE PES, OCT 2013. ISBN 978-1-4799-2984-9. DOI 10.1109/ISGTEurope.2013.6695431. pp. 21, 41, 87, 92, 94, 119, 122

A column generation based optimization method gives a large improvement in peak shaving performance over an earlier Lagrangian relaxation based method, and can address multiple objectives at the same time.

- [^{HAT}:4] TOERSCHE Hermen A, MOLDERINK A, HURINK JL & SMIT GJM. “Cascaded column generation for scalable predictive demand side management”. In *Energy-Con*. Cavtat, Croatia: IEEE, MAY 2014, pp. 1228–1235. ISBN 978-1-4799-2449-3. DOI 10.1109/ENERGYCON.2014.6850580. pp. 41, 97, 119

We apply a column generation procedure in a hierarchical configuration, address various practical problems, and quantify the convergence behaviour of the procedure.

- [^{HAT}:5] TOERSCHE Hermen A, HURINK JL & KONSMAN MJ. “Energy management with TRIANA on FPAI”. In *PowerTech*. Eindhoven, the Netherlands: IEEE PES, JUN 2015. DOI 10.1109/PTC.2015.7232650. p. 191

We show that the TRIANA concept adapts to the EF-Pi energy management platform.

- [^{HAT}:6] GERARDS Marco ET, TOERSCHE HA, HOOGSTEEN G, van der KLAUW T, HURINK JL & SMIT GJM. “Demand side management using profile steering”. In *PowerTech*. Eindhoven, the Netherlands: IEEE PES, JUN 2015. DOI 10.1109/PTC.2015.7232328. pp. 34, 38, 41, 45, 107, 111, 114, 234

DSM steering with linear prices promotes extreme behaviour. As an alternative, we propose a profile steering mechanism which specifies the desired profile directly.

- [^{HAT}:7] NYKAMP Stefan, MOLDERINK A, BAKKER V, TOERSCHE HA, HURINK JL & SMIT GJM. “Integration of heat pumps in distribution grids: economic motivation for grid control”. In *ISGT Europe*. Berlin, Germany: IEEE PES, OCT 2012. ISBN 978-1-4673-2596-7. DOI 10.1109/ISGTEurope.2012.6465605. pp. 257, 281, 282

Heat pump control gives substantial savings in distribution grid investment costs, especially in existing built-up areas.

- [^{HAT}:8] CLAESSEN Felix N, CLAESSENS BJ, HOMMELBERG MPF, MOLDERINK A, BAKKER V, TOERSCHE HA & van den BROEK MA. “Comparative analysis of tertiary control systems for smart grids using the Flex Street model”. In *Renewable energy* *69, Elsevier, SEP 2014, pp. 260–270. ISSN 0960-1481. DOI 10.1016/j.renene.2014.03.037. pp. 45, 117–119, 273, 280, 288, 290, 306

Comparing the performance of DSM methods is difficult, because there is no common test scenario. We compare TRIANA with Intelligator in the Flex Street scenario.

- [^{HAT}:9] ter BRAAK Timon D, TOERSCHE HA, KOKKELER ABJ & SMIT GJM. “Adaptive resource allocation for streaming applications”. In *International conference on embedded computer systems: architectures, modeling and simulation, IC-SAMOS 2011*. Samos, Greece: IEEE Circuits & Systems Society, JUL 2011, pp. 388–395. ISBN 978-1-4577-0800-8. DOI 10.1109/SAMOS.2011.6045489.

Run-time resource allocation enables the use of a multiprocessor system-on-chip with redundant hardware as a fault tolerant processing platform for streaming applications.

THIS THESIS

```
@phdthesis{toersche2016:phdthesis,
  author = {Toersche, Hermen A.},
  title = {Effective and efficient coordination
    of flexibility in smart grids},
  school = {University of Twente},
  address = {PO Box 217, 7500 AE Enschede, The Netherlands},
  year = {2016}, month = oct, day = {21},
  issn = {1381-3617}, number = {16-406},
  isbn = {978-90-365-4197-8},
  doi = {10.3990/1.9789036541978}
}
```


GLOSSARY

It has been estimated that, at the current rate of three letter acronym (TLA) allocation, we will run out by the end of September this year.

IETF, April 1st 2009 [115]

3G	third-generation mobile communication technology
AC	alternating current
AC	air conditioning
ADMM	alternating direction method of multipliers
ADP	approximate dynamic programming
ADR	automated DR
AGC	automatic generation control
AI	artificial intelligence
AMI	advanced metering infrastructure
AMR	automatic, or automated meter reading
API	application programming interface
APX	Amsterdam Power Exchange, electricity exchange market
ARM	Advanced RISC Machines, processor design company
BAU	business as usual
BEMS	building energy management system
BRP	balance responsible party
BTU	British thermal unit, 1 BTU $\approx$ 1055 J
C	programming language, de facto standard in embedded systems
C++	programming language
CAES	Computer Architecture for Embedded Systems
CAES	compressed air energy storage
CHP	combined heat and power
CO ₂	carbon dioxide
COP	coefficient of performance
CPLEX	optimization software, especially for MIP
CPP	critical peak pricing
CPR	critical peak pricing with rebate
CPU	central processing unit
CSA	control space adapter
CSP	concentrated solar power
CTIT	Centre for Telematics and Information Technology
CTMC	continuous time Markov chain

DAG	directed acyclic graph
DBU	daily battery use
DC	direct current
DER	distributed energy resource
DG	distributed generation
DHT	distributed hash table
DLC	direct load control
DMMP	Discrete Mathematics and Mathematical Programming
DOE	US Department of Energy
DP	dynamic programming
DR	demand response
DREAM	Dynamic Real-time control of Energy streAMs in buildings, STW project
DSM	demand side management
DSO	distribution system operator, also <i>utility</i>
DTMC	discrete time Markov chain
e-balance	FP7 research project
ECN	Energy research Centre of the Netherlands
ECU	engine control unit
EEMCS	Electrical Engineering, Mathematics and Computer Science
EFI	energy flexibility interface, part of EF-Pi
EF-Pi	Energy Flexibility Platform and interface
EIT	European Institute of Innovation and Technology
EMS	energy management system
ESCO	energy service company
EU	European Union
EV	electric vehicle
FFT	fast Fourier transform
FiT	feed-in tariff
Flex Street	simulation scenario, see Appendix B
FORTRAN	programming language, popular in HPC
GCD	greatest common divisor
GPRS	general packet radio service, legacy mobile data service, popular in utility automation
GPU	graphics processing unit
GUI	graphical UI
HEGRID	Hybrid Energy Grid Management, EIT digital research project
HPC	high performance computing
HRe	Hoog Rendement met elektriciteit; microCHP
HV	high voltage
HVAC	heating, ventilation and air conditioning
HVDC	high voltage DC
ICT	information and communication technology
ID	identity, or identifier
IDDP	iterative distributed dynamic programming
IEA	International Energy Agency
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force

IFFT	inverse FFT
ILP	integer linear programming, see MIP
Intelligator	hybrid smart grid control framework
IoE	internet of energy, distributed smart grid
IP	internet protocol, or intellectual property
IPM	interior point method
IR	intermediate representation
IRP	integrated resource planning
ISA	instruction set architecture
ISO	independent system operator ($\approx$ TSO)
Java	programming language
JIT	just-in-time
JVM	Java VM
KPI	key performance indicator, i.e. objective
LDC	load duration curve
LED	light emitting diode
LHS	left-hand side
LINEAR	Local Intelligent Networks and Energy Active Regions, Flemish smart grid research project
Linux	UNIX-like OS, popular among computer scientists and smartphone users
LLVM	Low Level Virtual Machine
LoRaWAN	Long Range Wide Area Network, low-power long-range protocol for IoT
LP	linear programming
LRU	least recently used
LUT	lookup table
LV	low voltage
MAPE	mean absolute percentage error
MAS	multi-agent system
MATLAB	scientific programming environment, <i>the language of technical computing</i>
microCHP	very small scale CHP
miniCHP	small scale CHP
MIP	mixed integer (linear) programming
MIQP	mixed integer quadratic programming
Modbus	serial communications protocol, originally for PLCs
MPC	model predictive control
MV	medium voltage
.NET	software framework, developed by Microsoft
NIMBY	not in my backyard
NLP	nonlinear programming
NREL	National Renewable Energy Laboratory, energy research institution
NTP	network time protocol
NumPy	numeric Python, “MATLAB for Python”
NWO	Nederlandse Organisatie voor Wetenschappelijk Onderzoek – Dutch Organisation for Scientific Research
OLTC	on load tap changer
OpenADR	ADR standard, popular in the US
OPF	optimal power flow

OR	operational research
OS	operating system
OSGi	Open Service Gateway initiative
P2G	power to gas
PAN	personal area network
PC	personal computer
PDF	probability density function
PEAS	performance, environment, actuators, sensors
PHEV	plug-in hybrid EV
PLC	power line communication
PLC	programmable logic controller
PMU	phasor measurement unit
PowerMatcher	auction-based smart grid control framework
Protocol Buffers	serialization format and library
PTR	peak time rebate, CPR
PV	photovoltaic solar panel
PWM	pulse width modulation
Python	programming language, as in <i>Monty Python</i>
QoS	quality of service
QP	quadratic programming
RAM	random access memory
renewables	see RES
RES	renewable energy source, also <i>renewables</i>
RHS	right-hand side
RISC	reduced instruction set computing
RMP	restricted master problem
RMS	root mean square
RPR	relative peak reduction
RTO	regional transmission organization ($\approx$ TSO)
RTP	real time pricing
RWE	major energy company based in Germany
SCOPF	security constrained OPF
SDM	supply and demand matching
SEER	seasonal energy efficiency ratio, describes cooling efficiency as the ratio of cooling BTUs to energy consumption
SES	smart energy system
SI	Système International d'unités
smart grid	smart grid
SoC	state of charge
SPF	seasonal performance factor, effective COP
STW	Stichting voor de Technische Wetenschappen – Dutch Technology Foundation, part of NWO
TKI	Topconsortia voor Kennis en Innovatie – Top consortia for Knowledge and Innovation, Dutch subsidy program
TL	fluorescent lighting, <i>tube luminescent</i>
TLA	three letter acronym
TLDR	too long, didn't read

TNO	Toegepast Natuurwetenschappelijk Onderzoek – Netherlands Organisation for Applied Scientific Research
TOU	time of use
TRIANA	decentralized smart grid coordination approach
TSO	transmission system operator
TV	television
UC	unit commitment
UI	user interface
UK	United Kingdom (of Great Britain and Northern Ireland)
URI	Uniform Resource Identifier
US	United States (of America)
V2G	vehicle to grid
VIOS	virtual infrastructure operating system, Dutch TKI research project
VITO	Vlaamse Instelling voor Technologisch Onderzoek – Flemish Institute for Technological Research
VM	virtual memory
VM	virtual machine
VPP	virtual power plant
WLAN	wireless local area network
WSN	wireless sensor network
x86	ISA of most Intel and AMD processors
XML	Extensible Markup Language
ZigBee	IEEE 802.15.4 based PAN standard

SYMBOLS

Common definitions

$\emptyset$	empty set
$\square$	null/undefined
$\mathbf{0}, \mathbf{1}$	vector of zeroes, ones
$\mathcal{O}(\cdot)$	complexity class (usually in time)
t	time, usually index of discrete interval in $\mathcal{T}$
$\mathcal{T}$	set of time intervals, usually all of length τ
n_t	number of time intervals ($ \mathcal{T} $)
τ	time interval length/duration, sometimes of nonuniform length
$\mathbf{x}, x(t)$	power demand vector over time (usually electricity in W, negative means supply)
$\mathbf{c}, c(t)$	pricing scheme vector over time (usually an electricity price vector in €/J or €/W τ , where τ indicates the time interval duration)
$\tau \mathbf{c}^\top \mathbf{x}$	commodity cost over considered time horizon (usually in €)
z	cost (usually in €) or objective value over considered horizon
k	iteration index
n_k	number of iterations
$\mathcal{U}(a, b)$	uniform distribution between and including a and b
$\mathcal{N}(\mu, \sigma)$	normal distribution with mean μ and variance σ^2
$\mathcal{B}(n, p)$	binomial distribution with n trials and success probability p
el, heat, th, gas	electricity, heat, thermal, gas (e.g. x_{el} for electric power)
ℓ	liter, linear

Chapter 3

Coordination in time

$\mathcal{x}$	node index (may e.g. refer to a neighbourhood, house, device)
$k_{\mathcal{x}}$	iteration within node $\mathcal{x}$
$\mathcal{I}_{\mathcal{x}}$	set of child nodes of a node $\mathcal{x}$
i, i'	node index for $\mathcal{I}_{\mathcal{x}}, \mathcal{I}_i$ (same type as $\mathcal{x}$)
$\mathbf{x}_{\mathcal{x}:1:2}$	path notation, refers e.g. to demand of second child of first child of $\mathcal{x}$
$\mathbf{x}_{\mathcal{x}}, \mathbf{x}_i$	demand of subtree of $\mathcal{x}$, node i
$\mathbf{x}_{\mathcal{x}}^{\text{down}}$	demand of descendants of $\mathcal{x}$

$\mathbf{x}_x^{\text{int}}$	demand of x itself without descendants
z_x^{sub}	internal cost over horizon of subtree of x
z_x^{down}	internal cost over horizon of descendants of x
z_x^{int}	internal cost over horizon of x itself without descendants
z_x	total cost over horizon of subtree of x , including z_x^{up}
$\mathbf{c}_i, \mathbf{c}_x$	(candidate) pricing scheme vector imposed on node i (for node x)
z_x^{up}	commodity cost over horizon perceived in node x resulting from $\mathbf{c}_x$
p, p_x	(candidate) result pattern of node x and its descendants, describes either an index into $\mathcal{P}_x$ or the pattern $\langle \mathbf{x}_{x,p_x}, z_{x,p_x}^{\text{sub}} \rangle$ itself
$\mathcal{P}_x, \mathcal{P}_i$	set of known possible patterns for node x , node i
π_x	(candidate) control strategy of node x corresponding to some p_x
$\mathbf{c}_x^\ell$	linear price vector component of commodity pricing scheme of node x
$\mathbf{c}_x^q$	quadratic weight vector component of commodity pricing scheme of node x
$\mathbf{x}_{x,c}$	demand of node x for a commodity $c \in \mathcal{C}$, e.g. $\mathbf{x}_{x,\text{gas}}$ for gas (usually implicitly $c = \text{el}$, i.e. electricity)
$\mathcal{C}$	set of commodities
$\mathbf{p}_x^{\text{goal}}$	goal demand profile over time
Section 3.3 <i>IDDP</i>	
$\mathbf{p}_x^{\text{upper}}, \mathbf{p}_x^{\text{lower}}$	upper, lower bound on demand over time
δ_x	$\mathbf{p}_x^{\text{upper}} = \mathbf{p}_x^{\text{goal}} + \delta_x, \mathbf{p}_x^{\text{lower}} = \mathbf{p}_x^{\text{goal}} - \delta_x$
$w_x^{\text{upper}}, w_x^{\text{lower}}$	weight of soft constraint on $\mathbf{p}_x^{\text{upper}}, \mathbf{p}_x^{\text{lower}}$
$\mathcal{T}_x^{\text{upper},(k)}$	set of time intervals that violate $\mathbf{p}_x^{\text{upper}}$ in iteration k
$\mathcal{T}_x^{\text{lower},(k)}$	set of time intervals that violate $\mathbf{p}_x^{\text{lower}}$ in iteration k
$\mathcal{T}_x^{\text{on},(k)}$	set of time intervals where device x is turned on in iteration k
$\mathcal{T}_x^{\text{off},(k)}$	set of time intervals where device x is turned off in iteration k
α	price update coefficient ($0 < \alpha < 1$, usually close to 1)
$p_x^{\text{up},(k+1)}(t)$	probability of price increase in iteration k for time interval t
$p_x^{\text{down},(k+1)}(t)$	probability of price decrease in iteration k for time interval t
$\beta_{x,t}^{(k)}$	scaling factor for probability of change in iteration k for time interval t
Section 3.4 <i>Column generation</i>	
$\bar{A}_i$	connecting constraint coefficients of subproblem i in (restricted) master problem
A_i	independent constraint coefficients of subproblem i
$\mathbf{x}_i$	(almost) independent variables of subproblem i
$\mathbf{b}_i$	right-hand side values of independent constraints of subproblem i

$\bar{\mathbf{b}}$	right-hand side values of connecting constraints
$\mathbf{c}_i$	objective weights corresponding to $\mathbf{x}_i$
$y_{i,p}^{(k)}$	assigned weight of pattern p of subproblem i in iteration k in restricted master problem
$z_{i,p}$	objective weight of pattern p in restricted master problem
$\bar{\mathbf{b}}_{i,p}$	coefficients of $y_{i,p}^{(k)}$ in restricted master problem, i.e. $\bar{\mathbf{b}}_{i,p} = \bar{A}_{i,p} \mathbf{x}_{i,p}$
δ_i	reduced cost of subproblem i
$\pi^{(k)}((3.14))$	dual solution/shadow prices/Lagrange multipliers in iteration k of constraints given e.g. in Equation 3.14
$\tilde{\pi}_i$	shadow price of mutex constraint on $\mathbf{y}_i$
$n_{k,\mathcal{N},\mathbf{I}}$	number of outer (integer rounding) iterations in node $\mathcal{N}$
$k_{\mathbf{I}}$	outer iteration index
$n_{k,\mathcal{N},\mathbf{r},k_{\mathbf{I}}}$	number of inner (relaxed) iterations in node $\mathcal{N}$ in outer iteration $k_{\mathbf{I}}$
$k_{\mathbf{r}}$	inner iteration index
$\mathbf{x}_i^{(k_{\mathbf{I}},k_{\mathbf{r}})}$	assignment e.g. to $\mathbf{x}_i$ in outer iteration $k_{\mathbf{I}}$ and inner iteration $k_{\mathbf{r}}$
$\mathcal{P}_{i,\text{cand}}^{(k_{\mathbf{I}},k_{\mathbf{r}}+1)}$	candidate pattern set produced for subproblem i before reduced cost test in iteration $(k_{\mathbf{I}}, k_{\mathbf{r}})$
$B^{(k_{\mathbf{I}},k_{\mathbf{r}})}$	bootstrap heuristic scaling factor
$n_{k,B}$	bootstrap heuristic iteration count
$x_{\mathcal{N},\text{max}}, w_{\text{max}}$	maximum of $\mathbf{x}_{\mathcal{N}}$, corresponding objective weight
$x_{\mathcal{N},\text{min}}, w_{\text{min}}$	minimum of $\mathbf{x}_{\mathcal{N}}$, corresponding objective weight (negative)
$x_{\mathcal{N},\text{mm}}, w_{\text{mm}}$	difference between maximum and minimum of $\mathbf{x}_{\mathcal{N}}$ and its weight
$d_{\mathcal{N}}(t), w_D$	difference between $x(t)$ and $x(t+1)$, corresponding objective weight
$d_{\mathcal{N},\text{max}}, w_{D,\text{max}}$	maximum difference between $x(t)$ and $x(t+1)$ and its weight
$z_{\mathcal{N},\mathbf{I}}^{(k_{\mathbf{I}})}$	objective cost of integer solution for $\mathcal{N}$ after iteration $k_{\mathbf{I}}$
$z_{\mathcal{N},\mathbf{r}}^{(k_{\mathbf{I}},k_{\mathbf{r}})}$	objective cost of relaxed solution for $\mathcal{N}$ in iteration $(k_{\mathbf{I}}, k_{\mathbf{r}})$
$z_{\mathcal{N},\mathbf{I},\text{est}}^{(k_{\mathbf{I}})}$	estimate of objective cost of integer solution for $\mathcal{N}$ after iteration $k_{\mathbf{I}}$
$z_{\mathcal{N},\mathbf{I},\text{round}}^{(k_{\mathbf{I}})}$	objective cost of integer solution for $\mathcal{N}$ after iteration $k_{\mathbf{I}}$ based on rounding of $\mathbf{y}_i$
$z_{\mathcal{N},\mathbf{r}}^{\text{top}}$	objective cost of relaxed solution for $\mathcal{N}$ without descendant cost
Section 3.5	<i>Profile steering</i>
$\mathbf{d}_{\mathcal{N}}$	difference between desired and current demand profile
$w_{p_{\text{goal}},\mathcal{N}}$	weight of difference between desired and current demand profile
$\alpha_{\mathcal{N}}$	square root compensating scaling factor
$\mathbf{x}_{i,\text{sel}}^{(k-1)}$	currently selected pattern in subproblem i e.g. for $\mathbf{x}_i$
$\tilde{z}_i^{(k)}$	improvement to subtree cost of $\mathcal{N}$ for pattern proposed by subproblem i (more negative is better)
$i^{(k)}$	winning subproblem in iteration k

Chapter 4	Coordination in space
$\mathcal{I}^d$	set of demand-side devices/agents (all devices)
$\mathcal{I}^s$	set of supply-side devices/agents (economic agent)
i	index for agent in $\mathcal{I}^d$
p, p_a, p_b	priority level ($\in \mathbb{R}$)
$x_{i,t}^d(p)$	demand bid of agent i at time t for priority p
$p_{i,t}^{d,\min}, p_{i,t}^{d,\max}$	lower, upper bound of “active” priority domain of i
$x_t^d(p)$	aggregate demand bidding curve at time t
$x_t^s(p)$	aggregate supply offer curve at time t
$p_t^{d,\min}, p_t^{d,\max}$	lower, upper bound of “active” domain of aggregate demand curve
$p_t^{s,\min}, p_t^{s,\max}$	lower, upper bound of “active” domain of aggregate supply curve
i^s	economic agent, only agent at the supply side ($\mathcal{I}^s = \{i^s\}$)
$x_{i^s,t}(p)$	supply offer curve of economic agent at time t
$p_{i^s,t}^{d,\min}, p_{i^s,t}^{d,\max}$	lower, upper bound of “active” priority domain of i^s
$p_t^*, \mathbf{p}^*$	clearing priority at time t
$\tilde{x}_t^s(p)$	aggregate supply offer curve at time t corrected for possible infeasible solutions
$x_t^d, \mathbf{x}^d$	realized demand profile over time, i.e. $x_t^d(p_t^*)$ (analog to $\mathbf{x}_N$)
$z^{\text{ext}}(\mathbf{x}^d)$	external costs for $\mathbf{x}^d$ (analog to z_N^{up})
z^{int}	internal costs of realization (analog to z_N^{sub})
$z(\mathbf{x}^d)$	total costs for $\mathbf{x}^d$ (analog to z_N)
$\pi, \mathbf{x}, \mathbf{x}^\pi$	supply offer strategy of economic agent
$\hat{x}_t^\pi(p)$	supply offer curve of i^s at time t under π
$p_t^\pi, \mathbf{p}^\pi$	clearing priority outcomes over time under π
$\mathbf{x}^d(\pi)$	demand profile over time under π
n_{ts}	number of represented time intervals
n_{ps}	number of represented priority levels
$t^s(t')$	represented time interval at index $t' \in \{1, \dots, n_{ts}\}$
$p^s(p')$	represented priority level at index $p' \in \{1, \dots, n_{ps}\}$
$\hat{x}_{t'}^s(p')$	supply offer value at time $t^s(t')$ and priority level $p^s(p')$
$\hat{x}_{t'}^s(p)$	supply offer curve of i^s at time $t^s(t')$
$\mathbf{\hat{x}}, \mathbf{\hat{x}}^\pi, \pi$	complete assignment of $\hat{x}_{t'}^s(p')$ ($n_{ts} \cdot n_{ps}$ values) that represents a supply offer strategy ($\pi \equiv \mathbf{\hat{x}}$)
$x_{i^s,t}^{s,\pi}(p)$	supply offer curve of i^s under $\pi/\mathbf{\hat{x}}^\pi$ at time t
$\text{interp}(\mathbf{a}, \mathbf{b}, a')$	interpolation from values $\mathbf{a}$ to $\mathbf{b}$ with clipping on $\min \mathbf{a}$ and $\max \mathbf{a}$
Ξ	realized scenario set
n_Ξ	number of scenarios in Ξ
ξ	index of realized scenario
p_ξ	weight of realized scenario

$\tilde{\Xi}$	estimated scenario set
$n_{\tilde{\Xi}}$	number of scenarios in $\tilde{\Xi}$
$\tilde{\xi}$	index of estimated scenario
$p_{\tilde{\xi}}$	weight of estimated scenario
$x_t^{d,\pi,\xi}(p)$	demand bidding curve under π at time t in scenario ξ
$x_t^{d,\pi,\tilde{\xi}}(p)$	demand bidding curve under π at time t in scenario $\tilde{\xi}$
$p_t^{\pi,\xi}, p_t^{\pi,\tilde{\xi}}$	clearing price under π at time t in scenario $\xi, \tilde{\xi}$
$\mathbf{x}^{d,\xi}(\pi), \mathbf{x}^{d,\tilde{\xi}}(\pi)$	demand profile over time under π in scenario $\xi, \tilde{\xi}$
$z(\pi), z(\mathbf{x}^{\pi})$	expected realization of total costs under π (weighted over Ξ)
$\tilde{z}(\pi), \tilde{z}(\mathbf{x}^{\pi})$	estimated total costs under π (weighted over $\tilde{\Xi}$)
$\tilde{\mathbf{c}}, \mathbf{c}^{\tilde{\xi}}$	estimated electricity prices
$\mathbf{x}^{*,\xi}, \mathbf{x}^{*,\tilde{\xi}}$	demand profile in reference optimal QP solution in scenario $\xi, \tilde{\xi}$
$z^{*,\xi}, z^{*,\tilde{\xi}}$	total costs in reference optimal QP solution in scenario $\xi, \tilde{\xi}$
$z^*, \tilde{z}^*$	total costs in reference optimal QP solution (weighted over $\Xi, \tilde{\Xi}$)
Section 4.3 <i>Metaheuristics</i>	
$\mathbf{x}^{s(k)}$	candidate assignment of $\pi/\mathbf{x}^{\pi}$ in iteration k
$\mathbf{x}_{\text{sel}}^{s(k-1)}$	current (usually best-known) assignment of $\pi/\mathbf{x}^{\pi}$ before iteration k
$d^{(k)}(t', p')$	proposed random change to $\mathbf{x}_{t'}^{s(k)}(p')$ in iteration k
$b^{(k)}(t', p')$	random outcome that determines whether $d^{(k)}(t', p')$ is accepted
p_+	probability of acceptance for $b^{(k)}(t', p')$
$p_{\text{accept}}^{(k)}$	probability of accepting a suboptimal solution (sim. annealing)
c_{accept}	scaling factor for $p_{\text{accept}}^{(k)}$
$d^{(k)}$	random outcome of whether a suboptimal solution is accepted
$\mathcal{X}_{is}^{(k)}$	population of assignments to $\mathbf{x}$
$n_{\mathcal{X}}$	maximum population size
$\mathcal{X}_{\text{gen}}^{(k)}$	candidate offspring parents from $\mathcal{X}_{is}^{(k)}$
$n_{\mathcal{X},\text{gen}}$	candidate offspring parents group size
$\mathbf{w}_{\mathcal{X},\text{gen}}^{(k)}$	random weights for candidate offspring parents (unscaled)
$\mathbf{w}_{\mathcal{X},\text{gen}}^{(k)}$	random weights of $\mathbf{w}_{\mathcal{X},\text{gen}}^{(k)}$ scaled to sum up to 1
$\mathbf{x}_{\text{gen},t'}^{s(k)}(p')$	candidate offspring base assignment
$\text{choice}(X)$	selects one member from X with uniform probability
$\mathbf{x}_{\mathcal{X},\text{max}}^{s(k)}, \mathbf{x}_{\mathcal{X},\text{min}}^{s(k)}$	worst, best solution in population in iteration k
Section 4.4 <i>Heat pump use case</i>	
$x_i(t)$	input power percentage of heat pump i ($\in [0, 1]$)
x_i^{max}	maximum electric power of heat pump
η_i	COP of heat pump

$x_i^+(t)$	electric power input of auxiliary resistance heating rod for i
$s_i(t)$	SoC of buffer for heat pump i after time t
$s_i^{\max}$	maximum SoC of buffer for heat pump i
$d_i(t), \mathbf{d}_i^\xi, \mathbf{d}_i^{\tilde{\xi}}$	average heat demand for i per time interval in scenario $\xi, \tilde{\xi}$
$x(t)$	aggregate electric demand of all heat pumps and heating rods at t
β	quadratic balancing coefficient in objective
z_s	conservative estimate of SoC value difference at end of simulation
s^*	target aggregate SoC level at end of simulation
$c_{\text{buy}}, c_{\text{sell}}$	price to buy, sell SoC difference at end of simulation
$\tilde{x}_{i,t}^{\max}, \tilde{x}_{i,t}^{\min}$	maximum, minimum demand bid power value at time t
$p_i^{\text{int}}(t)$	internal priority value/urgency at time t (determined by $s_i(t-1)$)
p^{ext}	rescaled priority value (determined by p)
$y_{i,t}^d(p^{\text{ext}})$	weight of $\tilde{x}_{i,t}^{\max}$ in demand bidding function ($\in [0, 1]$)
$\tilde{\mathbf{d}}_i$	expected average heat demand over time in lookahead window
$\mathcal{T}^{\text{ahead}}$	lookahead window for $\tilde{\mathbf{d}}_i$
t', t''	index for lookahead window
n_{fahead}	length of lookahead window
$l(x_{i,t})$	latest start time where forced to charge at full power when running heat pump i at a fraction $x_{i,t}$ of full power
$o(t')$	number of full power heat pump intervals to satisfy demand until t'
$\bar{o}(t')$	$o(t')$ corrected for periods with very high demand
$n(x_{i,t}, t')$	number of extra full power heat pump intervals to satisfy demand until t' when running heat pump i at a fraction $x_{i,t}$ of full power
$\bar{n}(x_{i,t}, t')$	$n(x_{i,t}, t')$ corrected for periods with very high demand
$p^{\text{bid},l}(x_{i,t})$	bid priority value for a given demand value of $x_{i,t}$ under l bidding
c_l	scaling factor for latest start time in bidding function
$m(x_{i,t}), m$	time in lookahead window that leads to highest average demand
n_x	number of considered demand value options
$p^{\text{bid},n/m}(x_{i,t})$	bid priority value for a given demand value of $x_{i,t}$ under n/m bidding
$\mathbf{c}_a, \mathbf{c}_b$	price vector alternatives (assigned both to $\mathbf{c}$ and $\tilde{\mathbf{c}}$)

Chapter 5 Coordination in practice

Section 5.4.2 Time Shiftable control space

i	index of device
j	index of <i>Time Shiftable</i> job
s	index of segment s in job j
$n_s, n_{s,i,j}$	number of segments in job j
$t_{i,j}^a$	arrival time of job j
$t_{i,j,s}^b$	begin time of segment s in job j

$t_{i,j,s}^c$	completion time of segment s in job j
$t_{i,j}^d$	deadline of job j
$\mathbf{x}_{i,j,s}$	demand profile of segment s in job j
$t^l, t_{i,j,s}^l$	duration of segment s in job j ($ \mathbf{x}_{i,j,s} $)
$t_{i,j,s}^{\text{gap}}, t_{i,j,s}^{\text{gap}}$	maximum off-time before segment s in job j in started job
$z_{\text{DP}}(t, s)$	cost-to-go at time t after having processed $s - 1$ segments
$z(x, t)$	commodity cost to consume x at time t
$z'(x, t)$	$z(x, t)$ windowed to $\mathcal{T}$
<i>Section 5.4.3 Buffer control space</i>	
s	fill level (SoC)
$ds/dt, p \rightarrow r$	fill rate
p, s, e	fill level function segment object
a, b	begin (left-hand side), end (right-hand side) of time interval
p_a, p_b	current fill level function segment object at begin (left-hand side), end (right-hand side) of time interval
n_s	number of segments/pieces in fill level function
$n_{s'}$	number of segments/pieces in combined fill level function
F	set of concurrent fill level functions
f	index of fill level function in F (sequence of segments)
f'	merged fill level function, result of Algorithm 5.1
q	priority queue variable
$\leftarrow$	insert in priority queue (queue on left-hand side), pop from priority queue (queue on right-hand side)
$\#$	concatenate lists
l_0	highest lower bound in F
u	lowest upper bound in F
$\text{ub}(f)$	upper bound of $\text{highest}(f)$
$\text{lb}(f)$	lower bound of $\text{lowest}(f)$
$\text{highest}(f)$	last fill level function segment in f
$\text{lowest}(f)$	first fill level function segment in f
$s \rightarrow l, s \rightarrow u$	lower, upper bound of segment s
$r, s \rightarrow r$	fill rate in segment s
$c, s \rightarrow c$	internal cost rate in segment s
$\mathbf{x}, s \rightarrow \mathbf{x}$	demand per commodity in segment s
$\mathbf{x}_{\text{max}}$	previous combined commodity consumption limit
$r_{\Delta}, c_{\Delta}, \mathbf{x}_{\Delta}$	difference between fill rate, internal cost rate, demand per commodity of segments within f
$r_0, c_0, \mathbf{x}_0$	fill rate, internal cost rate, demand per commodity at l_0

$s_{\downarrow}$	previous segment
f_{τ}	fill level function f applied for a specific time interval length τ (list of segments), as determined by Algorithm 5.2
d	reference to a (left-hand side) or b (right-hand side) of time interval
s_a, s_b, s_d	current fill level at left-hand side, right-hand side of time interval
t	time left within τ projection algorithm
t_a, t_b, t_d	time until left-hand side, right-hand side is reached at fill rate r
h_a	range of projected fill level function segment at left-hand side
$r', c', \mathbf{x}'$	change in projected fill rate, internal cost rate, commodity demand over s at left-hand side
$\text{next}(p)$	segment above p , or $\varnothing$ if $p = \text{highest}(f)$
n_b	number of discretized fill level points/bins
m	running mode (combined)
T	set of timers
n_{T}	number of timers
T_i	state of timer i
n_a	number of concurrent actuators

Appendix B

Flex Street

n_h	length of optimization horizon in time intervals
n_t	total length of simulation scenario
$\tilde{\mathbf{d}}, \mathbf{d}_i, \mathbf{d}, \mathbf{d}_i$	predicted, actual heat demand values in household i
$d_t(t')$	heat demand predicted at time t for time $t' \geq t$

Appendix B.3 Mathematical model of Flex Street

$\mathcal{I}$	set of all devices
$\mathcal{I}_{\varnothing}$	set of uncontrollable (profile) devices
$\mathbf{x}_{\varnothing, \mathcal{X}}$	demand vector of uncontrollable device $\mathcal{X}$
$\mathcal{I}_{\leftrightarrow}$	set of time shiftable devices
$J_{\leftrightarrow, \mathcal{X}}$	jobs of time shiftable device $\mathcal{X}$
$n_{J_{\leftrightarrow, \mathcal{X}}}$	number of jobs of time shiftable device $\mathcal{X}$
j	index of job in $J_{\leftrightarrow, \mathcal{X}}$
$t_{\leftrightarrow, \mathcal{X}, j, a}$	arrival time of job j
$t_{\leftrightarrow, \mathcal{X}, j, d}$	deadline of job j
$S_{\leftrightarrow, \mathcal{X}, j}$	set of segments of job j
$n_{S_{\leftrightarrow, \mathcal{X}, j}}$	number of segments of job j
$\mathbf{x}_{\leftrightarrow, \mathcal{X}, j, s}$	demand vector of segment s in job j of device $\mathcal{X}$
$t_{\leftrightarrow, \mathcal{X}, j, s, b}$	begin time decision for segment s
$t_{\leftrightarrow, \mathcal{X}, j, s, c}$	completion time of segment s
$\mathbf{y}_{\leftrightarrow, \mathcal{X}, j, s}$	vector of 0–1 indicators over time that indicates the value of $t_{\leftrightarrow, \mathcal{X}, j, s, b}$

$x_{\leftrightarrow, \mathcal{N}}(t)$	demand of $\mathcal{N}$ at time t
$\mathcal{I}_{\downarrow}$	set of (PH)EVs
$J_{\downarrow, \mathcal{N}}$	jobs of EV $\mathcal{N}$
$n_{J, \downarrow, \mathcal{N}}$	number of jobs of EV $\mathcal{N}$
j	index of job in $J_{\downarrow, \mathcal{N}}$
$t_{\downarrow, \mathcal{N}, j, a}$	arrival time of job j
$t_{\downarrow, \mathcal{N}, j, d}$	deadline of job j
$x_{\downarrow, \mathcal{N}, j}^{\max}$	maximum charge rate during job j
$E_{\downarrow, \mathcal{N}, j}$	charge amount during job j
$x_{\downarrow, \mathcal{N}}(t)$	demand of $\mathcal{N}$ at time t
$\mathcal{I}_{\text{bat}}$	set of batteries
$s_{\text{bat}, \mathcal{N}}(t)$	SoC of battery $\mathcal{N}$ after time interval t
$s_{\text{bat}, \mathcal{N}}^{\max}$	maximum SoC of battery $\mathcal{N}$
$x_{\text{bat}, \mathcal{N}}^{\min}$	maximum discharge rate of battery $\mathcal{N}$ (negative)
$x_{\text{bat}, \mathcal{N}}^{\max}$	maximum charge rate of battery $\mathcal{N}$ (positive)
$x_{\text{bat}, \mathcal{N}}(t)$	electric demand of battery $\mathcal{N}$ in time interval t (negative means supply)
$\eta_{\text{bat}, \mathcal{N}}$	input efficiency of battery $\mathcal{N}$
$\mathcal{I}_{\text{h}}$	set of heating systems
$\mathbf{x}_{\text{h}, \text{hp}, \mathcal{N}}$	electricity demand over time of heat pump in heating system $\mathcal{N}$
$x_{\text{h}, \text{hp}, \mathcal{N}}^{\max}$	maximum electricity demand of heat pump in heating system $\mathcal{N}$
$\mathbf{x}_{\text{h}, \text{aux}, \mathcal{N}}$	electricity demand over time of auxiliary resistance heating rod in heating system $\mathcal{N}$
$\tilde{\mathbf{d}}_{\text{h}, \mathcal{N}}, \mathbf{d}_{\text{h}, \mathcal{N}}$	predicted, actual heat demand over time for heating system $\mathcal{N}$
$s_{\text{h}, \mathcal{N}}(t)$	SoC of buffer for heating system $\mathcal{N}$ after time interval t
$s_{\text{h}, \mathcal{N}}^{\max}$	maximum SoC of buffer for heating system $\mathcal{N}$
$\eta_{\text{h}, \text{aux}, \mathcal{N}}$	COP of auxiliary resistance heating rod in heating system $\mathcal{N}$ ($= 1$)
$\eta_{\text{h}, \text{hp}, \mathcal{N}}$	COP of heat pump in heating system $\mathcal{N}$
$\eta_{\text{h}, \mathcal{N}, \tau}$	heat loss (exponential decay) of SoC in buffer of heating system $\mathcal{N}$ over time, expressed for a specific time interval length τ
$s_{\text{h}, \mathcal{N}}$	SoC step size in SoC-discretized variant of $\mathcal{N}$
$r_{\text{h}, \mathcal{N}}(t)$	number of SoC steps taken by $\mathcal{N}$ until the end of time interval t
$r_{\text{h}, \mathcal{N}, \text{ub}}(t)$	maximum number of SoC steps taken by $\mathcal{N}$ until the end of time interval t
$r_{\text{h}, \mathcal{N}, \text{lb}}(t)$	minimum number of SoC steps taken by $\mathcal{N}$ until the begin of time interval t

Appendix B.5 A case study based on Flex Street

n_{ctl}	number of controllable households
n_{WM}	number of controllable washing machines
n_{TD}	number of controllable tumble dryers

n_{DW}	number of controllable dishwashers
n_{RF}	number of controllable refrigerators
n_{FR}	number of controllable freezers
n_{EV}	number of controllable EVs
<i>Appendix B.6 Aggregate lower bound model</i>	
$\mathbf{P}_{\mathcal{X}}^{\text{ub}}$	maximum power demand value over time of device (group) $\mathcal{X}$
$\mathbf{P}_{\mathcal{X}}^{\text{lb}}$	minimum power demand value over time of device (group) $\mathcal{X}$
$\mathbf{E}_{\mathcal{X}}^{\text{ub}}$	maximum accumulated energy demand over time of device (group) $\mathcal{X}$ until end of time intervals
$\mathbf{E}_{\mathcal{X}}^{\text{lb}}$	minimum accumulated energy demand over time of device (group) $\mathcal{X}$ until begin of time intervals
$s_{\mathcal{X}}(t)$	accumulated energy demand of $\mathcal{X}$ after time interval t
$x_{\mathcal{X}}(t)$	electricity power demand of $\mathcal{X}$ in time interval t
$J_{\leftrightarrow, \mathcal{X}, t}$	potentially active jobs of time shiftable $\mathcal{X}$ at time t
<i>Appendix C Implementation details</i>	
n	number of subproblems
n_v	number of variables in demand vector
n_c	number of patterns to swap

Hermen A. Toersche

Effective and efficient coordination of flexibility in smart grids

ISBN 978-90-365-4197-8

9 789036 541978