


**Fractional
PROGRAMMING 
COOPERATIVE GAMES**

Xian Qiu

FRACTIONAL PROGRAMMING IN
COOPERATIVE GAMES

XIAN QIU

Composition of the graduation committee:

Chairman:	prof.dr.ir. A.J. Mouthaan	
Secretary:	prof.dr.ir. A.J. Mouthaan	Univ. Twente, EWI
Promotor:	prof.dr. M.J. Uetz	Univ. Twente, EWI
Ass. promotor:	dr. W. Kern	Univ. Twente, EWI
Referee:	dr. B. Manthey	Univ. Twente, EWI
Members:	dr. N. Litvak	Univ. Twente, EWI
	prof.dr. J.L. Hurink	Univ. Twente, EWI
	prof. U. Faigle	University of Cologne
	prof.dr. F.C.R. Spieksma	University of Leuven

CTIT

CTIT Ph.D. thesis series No. 13-254

Centre for Telematics and Information Technology

P.O. Box 217, 7500 AE, Enschede, The Netherlands

The work described in this thesis was performed at the Discrete Mathematics and Mathematical Programming group, Centre for Telematics and Information Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, The Netherlands.

Copyright © 2013 by Xian Qiu. All rights reserved. No part of this book may be reproduced or transmitted without prior written permission of the author.

Typeset with L^AT_EX. Printed by CPI-Wöhrmann Print Service - Zutphen, The Netherlands.

ISBN 978-90-365-0000-5/ISSN 1381-3617

DOI 10.3990./1.9789036500005

FRACTIONAL PROGRAMMING IN
COOPERATIVE GAMES

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof.dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended on Wednesday, 28 August, 2013
at 14:45

by

XIAN QIU

born on 29 April, 1985
in Hunan, China

This dissertation has been approved by:

Promotor: prof.dr. M.J. Uetz

Assistant promotor: dr. W. Kern

To my parents and my wife.

Abstract

Cooperation of individuals or institutions is often coupled with benefits that can be regarded as the monetary worth or outcome of the cooperation. Therefore, the problem naturally arises how to allocate the total outcome among institutions or individuals in a fair way. Such allocation problems are studied within cooperative game theory. A general idea is to find an allocation which guarantees a payoff no less than the earning of these players working without cooperation. This motivates a classical concept for “fair” division, the “core” allocation (Chapter 2) for all individuals. However, such core allocations are not always guaranteed in many practical and theoretical cases. Even if there exists a core allocation, finding such an allocation is often hard. For example, the bin packing game (Chapter 3) does not always admit a nonempty core and finding a core allocation for the bin packing game is an NP-hard problem. In case the core is empty, in this thesis, we adopt the tax model, *i.e.*, players can only keep a $(1 - \epsilon)$ fraction of their total earning if they work on their own, where ϵ is called the taxation rate. This is the general idea behind sales and tax, which is quite natural and acceptable. Based on this model, we aim at finding an ϵ -core allocation such that the taxation rate ϵ is as small as possible.

Motivated by this, it turns out that finding such a minimal ϵ is equivalent to finding the relative integrality gap of the integer linear programming formulation induced by the related characteristic function, where the relative integrality gap is defined by the optimal solution value of the ILP divided by the optimal solution value of its relaxation.

In particular, we consider the uniform bin packing game in Chapter 3 (based on papers [KQ12a] and [KQ13a]) and a more general

setting – the non-uniform bin packing game in Chapter 4 (based on [KQ12b]). A cooperative (uniform) bin packing game instance consists of k bins of capacity 1 each and n items of sizes a_1, a_2, \dots, a_n . In the non-uniform bin packing game, the bins may have different capacities. The value of a coalition of players is defined to be the maximum total size of items in the coalition that can be packed into the bins of the coalition. The main objective of study of this thesis is the relative integrality gaps of the uniform bin packing game and the non-uniform bin packing game.

Another part of this thesis studies approximation algorithms for facility location problems (Chapter 5, based on [KQ13b]). An instance of the facility location problem is defined by a set of cities and a set of possible facility locations. The task is to open a subset of the facilities and to assign each city to one of the opened facilities in such a way that the total cost, consisting of opening costs and connection costs is minimized. We will see how to use a factor-revealing LP to bound the approximation ratio of an approximation algorithm.

Acknowledgments

I feel so lucky to have spent five years (master plus PhD) life at the chair of Discrete Mathematics and Mathematical Programming, University of Twente. Having received countless concern and help from my colleagues, friends and parents, I really enjoyed my stay in the Netherlands. I am deeply grateful to all people who ever helped me.

In particular, it was my promotor Marc Uetz who offered me the chance of being a master and PhD candidate at the University of Twente. He is easy to approach and is always ready to help others. I particularly thank him for his efforts in getting financial support for the last year of my PhD.

Regarding my research, it was my daily supervisor Walter Kern who kept guiding me patiently in all aspects of my study, including choosing research topics, correcting proof details and giving presentations etc. The expert suggestions, technical advice and the motivational encouragement I received from him significantly improved my enthusiasm to academic research. We cooperated closely and had a lot of fun. I appreciate him very much for his endless help.

Also, I would express my thanks to Georg Still who is always prepared for helping others. The assistance I received from him as well as his friendly smile and fatherly love he showed to me will never go out of my memories. I thank Johann Hurink for his kind help during my study. I also thank Johann Hurink, Nelly Litvak and Bodo Manthey for their valuable comments to my thesis.

Many thanks to our secretary Marjo Mulder who arranged many wonderful events in our group and especially, I thank her for organizing a celebration party for my marriage. And of course, I was very

impressed and was moved for receiving sincerely wishes from all of my colleagues. It was really a enjoyable time of working with them.

Most important of all, I would deeply thank my parents and my grand parents who have always been supporting me unconditionally. And, I thank my wife Yuan Feng, who is also my colleague and will defend her PhD degree on the same day with me, for her trust and love. We were always together and will be.

Xian Qiu
Enschede, 2013

Contents

ABSTRACT	vii
ACKNOWLEDGMENTS	ix
ACRONYMS	xv
1 INTRODUCTION	1
1.1 Polynomial time	3
1.2 Approximation ratio	4
1.3 Graphs	5
1.4 Combinatorial optimization examples	7
1.4.1 Maximum weight matching	7
1.4.2 Multiple subset sum	10
1.4.3 Facility location	11
1.5 Problem statement	12
1.6 Outline of the thesis	14
2 ALLOCATION AND TAXES	15
2.1 Cooperative games and the core	16
2.2 The multiplicative ϵ -core	18
2.3 Operations research games	22
3 (UNIFORM) BIN PACKING GAME	23
3.1 Problem formulation	24
3.1.1 Integral and fractional packing	25
3.1.2 Non-emptiness of the ϵ -core	26
3.1.3 Reducing the problem size	28
3.2 Greedy selection	30

3.2.1	The basic idea	30
3.2.2	Proof of $1/3$ -core $\neq \emptyset$	32
3.3	The modified greedy selection	33
3.4	Large items: $\alpha_i > 1/3$	36
3.4.1	Fractional matching	36
3.4.2	Half-integrality	38
3.4.3	Tight lower bound	41
3.5	Matching plus greedy packing	43
3.5.1	The algorithm	44
3.5.2	The analysis	44
3.5.3	The relative integrality gap	46
3.6	Non-emptiness of the $1/4$ -core	47
3.6.1	Item packing	49
3.6.2	Greedy selection	51
3.6.3	The combination – set packing	53
3.7	Remarks and open problems	57
4	NON-UNIFORM BIN PACKING GAME	61
4.1	ILP formulation	62
4.2	Non-emptiness of the $1/2$ -core	63
4.3	Large items: $\alpha_i > 1/3$	66
4.3.1	On small items	67
4.3.2	Properties of the counterexample	68
4.3.3	Reversed greedy packing	71
4.4	Limiting case: $k \rightarrow \infty$	73
4.4.1	Restricting item sizes and bin sizes	73
4.4.2	Rounding items and bins	75
4.4.3	Dealing with small items	78
4.5	Remarks and open problems	81
5	FACILITY LOCATION PROBLEM	83
5.1	Introduction	83
5.2	Factor-revealing LP approach	85
5.2.1	The JMS algorithm	86
5.2.2	Factor-revealing LP	87
5.2.3	Scaling and augmentation	90

5.3	Analyzing the upper bound	92
5.3.1	Primal approach	93
5.3.2	Dual approach	99
5.4	Facility location games	103
5.5	Application to variants of the problem	107
5.5.1	The k-median problem	107
5.5.2	Facility location with arbitrary demands	110
5.5.3	Facility location with penalties	111
5.5.4	Fault-tolerant facility location	112
5.6	Remarks and open problems	113
6	SUMMARY	115
6.1	Relative integrality gaps	115
6.2	Facility location	117
	 BIBLIOGRAPHY	 119
	 PUBLICATIONS	 127

Acronyms

LP	Linear Program
ILP	Integer Linear Program
P	Class of decision problems that can be solved in polynomial time
NP	Class of decision problems that can be solved in polynomial time on a non-deterministic machine
PTAS	Polynomial Time Approximation Scheme
FPTAS	Fully Polynomial Time Approximation Scheme
Int.	Integral (solution)
Frac.	Fractional (solution)
OPT(Opt.)	Optimal (solution)
Supp	Support (set)
MST	Minimal Spanning Tree
MWM	Maximum Weight Matching
MSS	Multiple Subset Sum
FL	Facility Location

Introduction

Many problems of both practical and theoretical importance concern themselves with the choice of an optimal solution to achieve some objective. In economics, such an objective is often referred to as maximizing profits or minimizing costs. In these scenarios, decision makers must take a number of factors and requirements into consideration. In the mathematical formulation, these factors are *variables* and the requirements are *constraints*.

Given an *objective function* $f(x)$, where $x \in \mathbb{R}^n$ is the vector of decision variables, and given constraints $g_i(x) \geq 0$, for $i = 1, \dots, m$, $h_j(x) = 0$, for $j = 1, \dots, p$, the *optimization problem* (in terms of minimization problem) is indeed to find x solving

$$\begin{aligned} & \text{minimize } f(x) && (1.1) \\ & \text{subject to } g_i(x) \geq 0, \quad i = 1, \dots, m, \\ & \quad \quad \quad h_j(x) = 0, \quad j = 1, \dots, p. \end{aligned}$$

Techniques for solving such optimization problems have been widely studied in the literature. In particular, the problem is a *convex programming problem* when f is convex, g_i is concave and h is linear. The most convenient property of the convex programming problem is that local optimality implies global optimality. Besides, there are sufficient conditions for optimality, for instance the Karush-Kuhn-Tucker conditions [4] (which are not necessary in general).

For a large variety of practical optimization problems, they can even be formulated as *linear programming problems*, i.e., f , g_i and h_j are all

linear functions. The linear property brings us great convenience for computing optimal solutions for such problems. The famous *simplex algorithm* due to G.B. Dantzig [22] finds an optimal solution for linear programs. It has been shown that the optimal solution of a linear program lies in the set of extreme points of the polyhedron, defined by the linear inequalities of the linear program. The simplex algorithm starts with a feasible solution, and iteratively finds an extreme point which can improve the objective function value. The algorithm is generally regarded as quite efficient, though it may take exponential number of steps for solving some specially designed problems (cf. Klee and Minty [48]). Later, the *ellipsoid algorithm* for solving linear programming, developed by Shor, Khachiyan and Yudin (cf. textbook by Schrijver [65]), had been theoretically shown “efficient”, i.e., the number of steps in finding an optimal solution for a linear program is polynomial in the “size” of a problem. Afterwards, a more efficient algorithm which works well also in practice has been proposed by Karmarkar [41].

However, there are some problems whose solution set is discrete, i.e., the variables only take integer values. Solving such problems is very difficult in general, even if the variables are binary. For example, notorious problems such as SAT, Partition, Vertex Cover, TSP etc. are known as *hard* problems (cf. [59]). On the positive side, there are also some problems of this kind that can be easily solved, meaning that polynomial time algorithms have been found: MST, Matching, Shortest Path etc. are examples of “easy” problems. These optimization problems, having the feature that the solution set is discrete, are called *combinatorial optimization problems* (or *discrete optimization problems*). The computational complexity of combinatorial optimization problems largely depends on the properties of problems themselves. To study whether a problem is easy or hard is in the research interests of *computational complexity theory* (cf. Papadimitriou [60]), where the class of problems solvable in polynomial time is denoted as P and the class of NP contains all problems in P and many “hard” problems. In particular, the acronym NP stands for “nondeterministic polynomial

time". The class NP consists, roughly speaking, of all those decision problems (*i.e.*, answer "yes" or "no") with the property that with the help of a suitable "good guess" the "yes" instances can be verified in polynomial time.

1.1 POLYNOMIAL TIME

To characterize the *running time* of an algorithm, it is often hard to specify the exact number of running steps for each algorithm. We thus use *growth functions* to describe the asymptotic running time of an algorithm. Let $f, g : \mathbb{N} \rightarrow \mathbb{R}^+$. We define

$$O(g(n)) = \{f \mid \exists c > 0, n_0 \in \mathbb{N} \text{ such that } f(n) \leq c \cdot g(n) \forall n \geq n_0\},$$

$$\Omega(g(n)) = \{f \mid \exists c > 0, n_0 \in \mathbb{N} \text{ such that } f(n) \geq c \cdot g(n) \forall n \geq n_0\},$$

$$\Theta(g(n)) = \{f \mid f(n) \in O(g(n)) \text{ and } f(n) \in \Omega(g(n))\}.$$

For example, $(\frac{n}{2})^2 \log n^3 \in O(n^2 \log n)$, $n \log n \in \Omega(n)$, $2^{n+1} \in \Theta(2^n)$.

An *instance* I of an optimization problem can be specified as a pair (F, c) , where F is a set of feasible points and c is a mapping $c : F \rightarrow \mathbb{R}$. The objective (in terms of minimization problem) is to find an $f \in F$ such that $c(f) \leq c(g)$ for all $g \in F$. Such a point f is called a *globally optimal solution* to the given instance, or is simply called an *optimal solution* (if no confusion arises).

We need be careful to distinguish between a problem and an instance of a problem. Informally, a problem is a collection of instances, and an instance can be regarded as an example of the problem in which all parameters (input data) are given.

We say an algorithm is *polynomial* if it has a running time $O(p(|I|))$ for any instance I , where $|I|$ denotes the encoding length of I and p is a polynomial function.

1.2 APPROXIMATION RATIO

Among all decision problems in NP, the hardest problems are called NP-*complete* (and the corresponding optimization problems are called NP-hard). Many practical optimization problems that arise from industry and from our daily life turn out to be NP-hard. It is conjectured (also widely believed) that there does not exist efficient (polynomial time) algorithms for solving NP-hard problems [21]. Another perspective of looking at these problems is to find a “nearly” optimal solution, *i.e.*, a solution that is very close to optimal. The *approximation ratio* is a criterion which indicates how close the approximate solution is to the optimal solution. Roughly speaking, a good approximation algorithm will have approximation ratio close to 1.

However, the performance of an approximation algorithm may vary a lot according to different instances of a problem. For this reason, the approximation ratio of an algorithm is often referred to as its worst-case performance, *i.e.*, the performance guarantee applies for all instances of the problem.

Given an instance I of an optimization problem Π and an approximation algorithm A that solves Π , we denote by $A(I)$ and $OPT(I)$ the value of the approximate solution found by A and the value of an optimal solution respectively for instance I . The performance of the algorithm A is often measured by its *approximation ratio*.

An algorithm A for a minimization problem is an α -*approximation algorithm* with $\alpha \geq 1$ if it computes for every instance I in polynomial time a solution such that $A(I) \leq \alpha \cdot OPT(I)$.

An algorithm A for a maximization problem is an α -*approximation algorithm* with $\alpha \leq 1$ if it computes for every instance I in polynomial time a solution such that $A(I) \geq \alpha \cdot OPT(I)$.

Approximation algorithms can be regarded as a balance between efficiency and exactness. A good approximation algorithm may lose some exactness but may gain a great efficiency, which is in particular useful in real-time scenarios where efficiency is much more important than optimality. There are many successful approaches for designing

approximation algorithms, *e.g.* greedy, local search, primal-dual, LP-rounding etc. (*cf.* textbook by Williamson and Shmoys [73]). We especially concern ourselves with the LP-rounding technique for NP-hard problems. As we introduced before, linear programming can be solved efficiently. Naturally, the LP-rounding technique first relaxes integer variables, then finds an optimal solution for the corresponding (relaxed) linear program. Besides, the optimal objective function value of the linear program is expected to be close to that of the original integer program (otherwise this relaxation does not help much). Based on the given optimal solution, with fractional components, one wants to round them into integers, such that the rounded solution is feasible to the original problem. We consider suitable LP-relaxations, *i.e.* “fractional versions” of certain combinatorial problems and call these the (corresponding) *fractional programming problems*.

In the following sections, we give an extended introduction to the background of our fractional programming problems that underlie this thesis. In Section 1.3 below, we give some basic definitions in graph theory that will be used in upcoming chapters. In Section 1.4, we present some examples of combinatorial optimization problems as well as their motivations. In Section 1.5, we state the main problems that underlie this thesis. Finally, in section (1.6), we outline the main topics of each chapter.

1.3 GRAPHS

Combinatorial optimization problems can often be represented in the language of graph theory. In this section, we introduce some basic definitions about graphs.

An undirected *graph* G is a pair $G = (V, E)$, where V is a finite set of *nodes/vertices* and E is a set of *edges* that consists of pairs of vertices in V . For convenience of notation, given a graph G , we often write $V(G)$, $E(G)$ as the set of nodes and edges respectively. Each edge $e \in E$ is associated with an *unordered* pair $(u, v) \in V \times V$, which are called the

endpoints of e . If two edges have the same endpoints, then they are called *parallel* edges. An edge whose two endpoints are the same is a *loop*. A graph that neither has parallel edges nor loops is said to be *simple*. Note that in a simple graph every edge $e = (u, v) \in E$ is uniquely identified by its endpoints u and v . We always assume that undirected graphs are simple.

A subgraph H of G is a graph such that $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$, and each $e \in E(H)$ has the same ends in H as in G .

Given a subset $V' \subseteq V(G)$ of nodes, the *vertex induced subgraph* (or simply called induced subgraph) H has

$$V(H) = V' \text{ and } E(H) = \{(u, v) \mid u, v \in V', (u, v) \in E(G)\}.$$

Given a subset $E' \subseteq E(G)$ of edges, we can similarly define the *edge induced graph* H , containing E' as the edge set and the nodes of all endpoints of edges in E' .

Given $E' \subseteq E(G)$, $G \setminus E'$ refers to the subgraph H of G that we obtain if we delete all edges in E' from G , *i.e.*, $V(H) = V(G)$ and $E(H) = E(G) \setminus E'$. Similarly, given a subset $V' \subseteq V(G)$, $G \setminus V'$ refers to the subgraph H of G that we obtain if we delete all nodes in V' and its incident edges from G , *i.e.*,

$$V(H) = V(G) \setminus V' \text{ and } E(H) = E(G) \setminus \{(u, v) \in E(G) \mid u, v \in V'\}.$$

A subgraph is called *spanning* if it contains all nodes of G .

A *path* P in an undirected graph G is a sequence $P = \langle v_1, \dots, v_k \rangle$ of nodes such that $e_i = (v_i, v_{i+1})$ is an edge of G . We say that P is a path *from* v_1 *to* v_k or a v_1, v_k -*path* and v_1, v_k are the *start vertex* and end vertex respectively. We often refer to the *length* of a path P as the total number of edges in P . P is *simple* if all v_i ($1 \leq i \leq k$) are distinct. A *cycle* is a path for which the start vertex coincides with the end vertex and it is a *simple cycle* if all of its vertices are distinct (except for the start and end vertex). A graph is said to be *acyclic* if it does not contain a cycle.

A *connected component* $C \subseteq V$ of an undirected graph G is a subset of nodes such that for every two nodes $u, v \in C$, there is a u, v -path

in G . A graph G is said to be *connected* if for every two nodes $u, v \in V$ there is a u, v -path in G . A connected subgraph that does not contain a cycle is called a *tree*. A *spanning tree* T of G is a tree that contains all nodes of G .

1.4 COMBINATORIAL OPTIMIZATION EXAMPLES

Many combinatorial optimization problems can be formulated as (integer) linear programming. In the following, we present some specific examples of combinatorial optimization problems that arise in practice. The reader will see later that these problems will be dealt in Chapters 3, 4, and 5. However, instead of finding optimal solutions for these problems, we focus on the relative integrality gaps and the approximate core allocations (*cf.* Chapter 2) for their corresponding related games.

1.4.1 *Maximum weight matching*

Assume that n players are going to participate in a competition which is played in teams of two players. For a team, consisting of players i and j , we have an evaluation w_{ij} for its capability, which is a measure of the chance (or the utility) for this team to win the competition. The capability is affected, for example, by the preferences of players and the ability of each player and so on. Note that for some pair of players, they may not want to cooperate because they dislike each other. The coach wants to assign players to teams such that the total winning chance of all teams is maximized.

We create a graph $G = (V, E)$, where the node set V represents n players and the edge set E represents all possible teams of two players. For each edge $e \in E$, we assign weight w_e which stands for the winning chance (or the utility) of the team. Note that a player can only stay in one team in a competition. The formation of these teams is called a *matching*.

Formally, a *matching* of a graph G is a set of edges such that no two edges share a vertex, and the total weight of a matching M is the sum of the weights of the matching, *i.e.* $\sum_{e \in M} w_e$. Thus, the above problem can be formulated as the *maximum weight matching* problem (see Figure 1.1 for an example instance):

MAXIMUM WEIGHT MATCHING (MWM)

Given an undirected graph $G = (V, E)$ and edge weights $w : E \rightarrow \mathbb{R}^+$, find a matching with the maximum total weight.

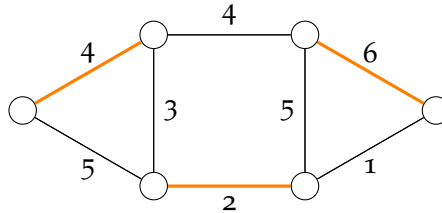


Figure 1.1: A maximum weight matching.

In the following, we will derive an integer linear program for the maximum weight matching problem. Denote by M a feasible matching and define variables

$$x_e = \begin{cases} 1, & \text{if } e \in M, \\ 0, & \text{otherwise.} \end{cases}$$

Thus, the MWM problem can be formulated as follows

$$\begin{aligned} & \text{maximize} && \sum_{e \in E} w_e x_e && (1.2) \\ & \text{subject to} && \sum_{e \ni v} x_e \leq 1, && \forall v \in V, \\ & && x_e \in \{0, 1\}, && \forall e \in E. \end{aligned}$$

MWM can be solved in polynomial time, e.g. the well-known blossom algorithm due to Jack Edmonds [24] finds a maximum weight matching in $O(|V|^4)$. Concerning the fractional version of this problem, the associated linear program of (1.2) obtained by relaxing $x_e \in \{0, 1\}$ to $x_e \geq 0$ has an optimal solution that is *half integral* (cf. Lovász and Plummer [55]).

Theorem 1.1 ([55]). *An optimal basic feasible solution of the associated linear program of (1.2) takes values in $\{0, 1/2, 1\}$.*

A feasible solution of the associated linear program of (1.2) (by relaxing $x_e \in [0, 1]$) is called a *fractional matching*. The following instance (Figure 1.2) shows that the maximum weight fractional matching can be strictly larger than the maximum weight integral matching.

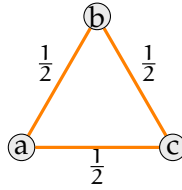


Figure 1.2: A maximum weight fractional matching.

In the above example, each edge has a weight 1. The fractional matching contains edges (a, b) , (b, c) , (c, a) , with a fraction $1/2$ each. Thus, the maximum fractional matching has a total weight $3/2$, which is larger than the maximum weight integral matching of value 1.

Given a maximum weight matching instance, let us denote v_{MWM} , v'_{MWM} as the total weight of the maximum integral and fractional matching respectively. Based on Theorem 1.1, Faigle and Kern [27] proved the following.

Theorem 1.2 ([27]). *Assume G contains no cycle of length smaller than k . Then $v_{\text{MWM}} \geq (1 - \frac{1}{k})v'_{\text{MWM}}$.*

A tight example is the cardinality matching (i.e. $w_e = 1$ for all $e \in E$) in an odd circuit of length k . However, if we consider a special case,

say G is *bipartite*, i.e., G contains no circuit of odd length ≥ 3 , then the integrality gap is 0 (cf. textbook by Cook *et al.* [20]).

Theorem 1.3 (Birkhoff's Theorem). *If G is bipartite, then $v_{MWM} = v'_{MWM}$.*

1.4.2 Multiple subset sum

An express company has k trucks of (uniform) capacity C each. A customer wants to transport n items of weight (or size) a_1, \dots, a_n to another city. If an item gets transported, the customer will pay the transport cost according to the weight of the item. W.l.o.g., assume the transport cost is proportional to the weight of an item. To maximize the profit, the express company wants to pack items to k trucks such that the total weight of packed items per truck does not exceed the capacity C and the total weight of all trucks is maximized.

We describe this problem in a "bin packing" version.

MULTIPLE SUBSET SUM (MSS)

Given k bins of capacity C each and n items of sizes a_1, \dots, a_n . The goal is to find an assignment of the items to the bins such that the total size of packed items is maximized (note that the total size of packed items in each bin can not exceed C).

We call a set F of items having a total value at most C a *feasible set* and let $a_F := \sum_{i \in F} a_i$. Denote by \mathcal{F} the collection of all feasible sets. So the multiple subset sum problem is to find the maximum value of k disjoint feasible sets. Define variables $x_F \in \{0, 1\}$, for all $F \in \mathcal{F}$, indicating whether F is assigned to a bin or not. Thus, the multiple subset sum problem can be formulated as below.

$$\begin{aligned} & \text{maximize} && \sum_{F \in \mathcal{F}} a_F x_F && (1.3) \\ & \text{subject to} && \sum_{F \ni i} x_F \leq 1, && i = 1, \dots, n, \end{aligned}$$

$$\sum_{F \in \mathcal{F}} x_F \leq k,$$

$$x_F \in \{0, 1\}, \quad \forall F \in \mathcal{F}.$$

This problem is strongly NP-hard as it is a generalization of the NP-complete 3-Partition problem (cf. [31]). Thus, we do not expect to find an efficient algorithm to solve it. As we will see afterwards (in Chapter 3) multiple subset sum contains all instances of maximum weight matching with restricted cardinality k . The structure of an optimal solution in the associated linear program of (1.3) is more complicated than that of a maximum weight fractional matching. We even do not know when the half integrality holds in this case. A result due to Woeginger [74] stated a lower bound on the relative integrality gap $v_{\text{MSS}}/v'_{\text{MSS}}$ (cf. Section 1.5), where as before, we denote by v_{MSS} , v'_{MSS} the optimal values of (1.3) and the associated linear program respectively.

Theorem 1.4 ([74]). $v_{\text{MSS}} \geq \frac{2}{3}v'_{\text{MSS}}$.

1.4.3 Facility location

In the facility location (FL) problem, we have a set \mathcal{C} of *cities* and a set \mathcal{F} of *facilities*. Cities are in need of certain supply (e.g. electricity, water, gas etc.) which can be provided by connecting them to facilities in \mathcal{F} . For all cities, one has to decide which facilities to open and then each city must be assigned to an opened facility. The cost of opening a facility $i \in \mathcal{F}$ is denoted by f_i . For any city $j \in \mathcal{C}$, there is a corresponding connection cost c_{ij} if it is assigned to the opened facility i . The goal is to minimize the total cost (opening costs plus connection cost) such that each city is assigned to an open facility.

FACILITY LOCATION (FL)

Given a set of facilities \mathcal{F} and a set of cities \mathcal{C} , an opening cost function $f : \mathcal{F} \rightarrow \mathbb{R}^+$ and a cost function $c : \mathcal{F} \times \mathcal{C} \rightarrow \mathbb{R}^+$. The goal is to assign each city to exactly one facility such that the total cost (opening costs plus connection costs) is minimized.

For all facilities $i \in \mathcal{F}$ and cities $j \in \mathcal{C}$, we define variables x_{ij} , y_i as below:

$$x_{ij} = \begin{cases} 1, & \text{if city } j \text{ is connected to facility } i, \\ 0, & \text{otherwise,} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{if facility } i \text{ is opened,} \\ 0, & \text{otherwise.} \end{cases}$$

The integer linear program of facility location problem can be written as follows.

$$\begin{aligned} & \text{minimize} && \sum_{i \in \mathcal{F}} f_i y_i + \sum_{i \in \mathcal{F}, j \in \mathcal{C}} c_{ij} x_{ij} && (1.4) \\ & \text{subject to} && \sum_{i \in \mathcal{F}} x_{ij} = 1, \quad \forall j \in \mathcal{C}, \\ & && x_{ij} \leq y_i, \quad \forall i \in \mathcal{F}, j \in \mathcal{C}, \\ & && x_{ij}, y_i \in \{0, 1\}, \quad \forall i \in \mathcal{F}, j \in \mathcal{C}. \end{aligned}$$

The facility location problem is a basic problem in combinatorial optimization and has many applications. Regarding the computational complexity, it is known to be NP-hard (even if all opening costs are the same, *cf.* [34]). We will mainly study its approximation algorithms later in Chapter 5.

1.5 PROBLEM STATEMENT

We have seen in Section 1.4 that optimization problems such as maximum weight matching in bipartite graphs have the property that

the value of an optimal solution of ILP coincides with that of its LP-relaxation. In the multiple subset sum problem, even though this integrality property does not hold, there is still some bound which estimates the deviation of the fractional optimum from the integral optimum for any instance. Thus, we are interested in the “gap” between the integral optimum and its fractional optimum for a given combinatorial optimization problem.

Given an instance I of the integer linear programming problem, we denote by $v(I), v'(I)$ the optimal value of I and the optimal value of the associated LP-relaxation (fractional version) respectively. Define the *integrality gap* as the absolute value of the difference between $v(I)$ and $v'(I)$, *i.e.*

$$\text{gap}(I) = |v(I) - v'(I)|.$$

Similarly, we define the *relative integrality gap* as

$$\text{gap}_{\text{rel}}(I) = \frac{v(I)}{v'(I)}.$$

Note that $v(I)/v'(I)$ is often referred to as the “integrality gap” in the literature (without the term – relative). In this thesis, we call this relative integrality gap so as to distinguish it from the absolute gap $|v'(I) - v(I)|$. (As $|v'(I) - v(I)|$ is bounded for some instances and we need to use this bound to analyze the relative integrality gap.)

In this thesis, we concern ourselves with the relative integrality gap of integer linear programming formulations for bin packing games as well as the facility location problem. As we have introduced at the beginning, LP rounding is a very useful heuristic for designing approximation algorithms. Given an optimization problem, we analyze the upper bound (or lower bound) of the relative integrality gap for all instances. If the relative integrality gap has a constant bound, then there is a great hope that we can find an LP rounding algorithm having a constant approximation ratio as well.

Besides, we consider optimization problems in a game theoretic context: The optimal value of an optimization problem is interpreted as the total gain (or total cost) of *players*. The question is how to allocate

the total earning among all players in a fair way. We will see in Chapter 2 that studying the relative integrality gap also has implications in such allocation problems.

1.6 OUTLINE OF THE THESIS

We have introduced the mathematical programming of some combinatorial optimization problems and have presented some results on the (relative) integrality gap in the introductory chapter. In Chapter 2, we consider allocation problems in the framework of cooperative game theory and introduce the (multiplicative) ϵ -core allocation for cooperative operations research games. We will see that studying the ϵ -core allocation of a cooperative game is equivalent to studying the relative integrality gap of the related integer linear program. Chapter 3 is based on papers [KQ12a] and [KQ13a], where the approximate core allocation for the uniform bin packing game is studied. In this chapter, we propose new packing heuristics, *e.g.* greedy selection, greedy packing and set packing, which will be used to analyze the relative integrality gap. Chapter 4 is based on paper [KQ12b], which extends the bin packing game to the non-uniform case, *i.e.*, bins are allowed to have different capacities. The packing heuristics used in Chapter 3 fail in the non-uniform case and completely new techniques for analyzing this more general scenario are investigated. Chapter 5 is based on [KQ13b]. We consider the (metric) facility location problem and investigate its approximation algorithms as well as the approximate core allocation for its game theoretical version. Finally, conclusions of our research are summarized in Chapter 6.

Allocation and taxes

A cooperative game is concerned primarily with groups of players who coordinate their actions and pool their winnings. Consequently, one of the problems here is to fairly divide the earnings among the members of the groups, so that every player is still willing to cooperate. Let N be a non-empty finite set of players. A subset $S \subseteq N$ is referred to as a *coalition* and the set N is called the *grand coalition*.

The problem of finding a fair allocation for players of the grand coalition is a central question in cooperative game theory (cf. [61]). In this setting, “fairness” must be considered in the first place. There are many ways of defining fairness, one famous concept is the *core* of a game, which guarantees that any coalition $S \subseteq N$ of players does not have an incentive to work on their own under allocation rules that are in the core.

However, for many cooperative games, a core allocation may not always exist. This means that no matter how we allocate, there always exists some coalition $S \subseteq N$ of players such that they can earn more on their own than what they get (when they cooperate). In this case, one either needs to pass to completely different solution concepts, e.g., the Shapley value [67], nucleolus [64] etc. or to relax the conditions of the core. In Section 2.2, we consider the multiplicative ϵ -core, proposed by Faigle and Kern [27]. The multiplicative ϵ -core can be seen as an ϵ -approximation to the core and is motivated by taxation in our daily life, where the total earning of players is taxed by some tax rate ϵ . The advantage of this definition is that we can always find an ap-

appropriate (as small as possible) ϵ such that the (multiplicative) ϵ -core is nonempty.

2.1 COOPERATIVE GAMES AND THE CORE

A cooperative game in characteristic function form is a pair $\langle N, v \rangle$ consisting of the player set N and the characteristic function (value function) $v : 2^N \rightarrow \mathbb{R}$ with $v(\emptyset) = 0$. (We denote by 2^N the set of all coalitions, *i.e.* the collection of all subsets of N .) Commonly the player set is given by $N = \{1, \dots, n\}$ and for each $S \subseteq N$ we denote by $|S|$ the number of elements of S .

A cooperative game $\langle N, v \rangle$ is called *additive* if $v(S \cup T) = v(S) + v(T)$ for all $S, T \subseteq N$ with $S \cap T = \emptyset$. For an additive game, we have $v(S) = \sum_{i \in S} v(\{i\})$ for all $S \subseteq N$, which is to say, in this case coalition S gets no extra profit if players cooperate, compared to working individually. Thus, additive games are trivial and the games we will consider are superadditive (or subadditive if v is regarded as cost, *cf.* below):

A cooperative game $\langle N, v \rangle$ is called *superadditive* if $v(S \cup T) \geq v(S) + v(T)$ for all $S, T \subseteq N$ with $S \cap T = \emptyset$.

Let $x = (x_i)_{i \in N} \in \mathbb{R}^N$ be an *allocation vector*, where x_i indicates the payoff of player i . For any coalition $S \subseteq N$, we denote by $x(S) := \sum_{i \in S} x_i$ the total payoff of players in S . The *core* is the collection of allocation vectors that satisfy

- (i) $x(N) = v(N)$,
- (ii) $x(S) \geq v(S)$ for all $S \subseteq N$.

The first condition is called the *efficiency condition*, which is quite natural and means that the total payoff is equal to the total earning of the grand coalition. The second condition says that for a coalition S , the total payoff can be no less than what the coalition could gain on its own.

In case of cost allocation, where the characteristic function value is interpreted as the total cost (of some coalition), players want to pay

the cost as little as possible. Thus, the second condition of the core should be $x(S) \leq c(S)$, where $c(S)$ denotes the total cost of S and c denotes the characteristic function of the game. A game is called *subadditive* if $c(S \cup T) \leq c(S) + c(T)$ for all $S, T \subseteq N$ with $S \cap T = \emptyset$.

A game is *balanced* if the core is nonempty for all instances. For ease of description, we use the word “always” indicating that some statement is true “for all instances”. So a balanced game can restated as a game whose core is always nonempty. The following theorem characterizes the non-emptiness of the core.

Theorem 2.1 ([68]). *A game (N, v) is balanced if and only if there is a function $\lambda : 2^N \rightarrow \mathbb{R}^+$ such that $\sum_{S \subseteq N} \lambda(S)v(S) = v(N)$ and $\sum_{S \ni i} \lambda(S) \leq 1$ for all $i \in N$.*

The proof of this lemma follows from linear programming duality, where $\lambda(S)$ are actually dual variables. A more general result on the balancedness will be presented in Section 2.2. In the following, we give examples of balanced games and unbalanced games.

MINIMAL COST SPANNING TREE GAME. Consider a network that is composed of a common supplier connected to n (geographically separated) users by a minimum cost spanning tree graph. An example of this situation is to build a network connecting n offices to the common supplier at a minimum cost. Here we concern how to allocate the total cost among n offices: Let $N = \{1, 2, \dots, n\}$ be the set of players and let 0 be the common supplier. The cost function $c(S)$ for $S \subseteq N$ is the cost of the minimal spanning tree on $S \cup \{0\}$. It has been shown by Granot and Huberman [33] that the minimal spanning tree game always has a nonempty core. In other words, the minimal spanning tree game is balanced.

BIN PACKING GAME. A (uniform) bin packing game N consists of k bins of capacity 1 each and n items of sizes a_1, a_2, \dots, a_n . The value function $v(S)$ for $S \subseteq N$ (containing bins and items) is the maximum total size of items of S that can be packed into the bins of S . In other

words, $v(S)$ is the optimum solution value of the multiple subset sum problem (cf. Section 1.4.2) w.r.t. item set S .

In particular, we consider a bin packing game instance consisting of 2 bins and 4 items of sizes $1/2, 1/2, 1/2, 2/3$. By symmetry, in a fair allocation each bin must be paid the same and all items of the same size must be paid the same as well. Let us denote by x, y the payoffs of items of size $1/2$ and $2/3$ and denote by z the payoff of each bin. First we observe that by packing two items of size $1/2$ to a bin and packing the item of size $2/3$ to another bin results in a maximum packing value $v(N) = 5/3$. Thus, by the efficiency condition, we have

$$3x + y + 2z = v(N) = \frac{5}{3}.$$

Again, if we only consider items that are packed in an optimum packing (and two bins), their total payoff should be no less than $v(N)$, we have

$$2x + y + 2z \geq \frac{5}{3}.$$

Combining the above two equations implies $x = 0$ and $z \leq 5/6$. As any two items of size $1/2$ can be packed to a bin, we have $2x + z \geq 1$, implying $z \geq 1$, a contradiction. Thus, this example shows that the uniform bin packing game is not balanced.

2.2 THE MULTIPLICATIVE ϵ -CORE

For unbalanced games $\langle N, v \rangle$, where v is regarded as the total earning (w.l.o.g., assume $v \geq 0$), we define the multiplicative ϵ -core as follows. Given $\epsilon \in [0, 1]$, we say an allocation vector x is in the ϵ -core if

- (i) $x(N) = v(N)$,
- (ii') $x(S) \geq (1 - \epsilon)v(S)$ for all $S \subseteq N$.

Note that if $\epsilon = 0$, the definition coincides with the core. We can interpret ϵ in condition (ii') as taxation rate in the sense that the players in coalition $S \subseteq N$ may keep only a $(1 - \epsilon)$ fraction of their total earning

on their own. This is a usual idea behind tax and, therefore, appears to be quite realistic and acceptable for the players.

A game with non-empty ϵ -core for all instances is called ϵ -balanced. In this sense, the ϵ -core provides an ϵ -approximation to balancedness. It can easily be seen that the 1-core is always non-empty for all games (with $v \geq 0$). In general, we seek to find a taxation rate ϵ as small as possible such that the ϵ -core is non-empty for a given class of games. Faigle and Kern [27] studied the (uniform) bin packing game and provided a necessary and sufficient condition for the non-emptiness of the ϵ -core, based on the linear programming description of the core (cf. Lemma 2.2). We extend this result to a more general class – superadditive games.

We consider superadditive games with nonnegative characteristic functions. The corresponding “core allocation problem” is

$$\begin{aligned} & \text{minimize } x(N) && (2.1) \\ & \text{subject to } x(S) \geq v(S), \quad \forall S \subseteq N. \end{aligned}$$

Note that $x \geq 0$ is implied as v is nonnegative. Its dual problem can therefore be written as

$$\begin{aligned} & \text{maximize } \sum_{S \subseteq N} v(S) y_S && (2.2) \\ & \text{subject to } \sum_{S \ni i} y_S \leq 1, \quad \forall i \in N, \\ & \quad y_S \geq 0, \quad \forall S \subseteq N. \end{aligned}$$

Note that the corresponding integer linear program

$$\begin{aligned} & \text{maximize } \sum_{S \subseteq N} v(S) y_S && (2.3) \\ & \text{subject to } \sum_{S \ni i} y_S \leq 1, \quad \forall i \in N, \\ & \quad y_S \in \{0, 1\}, \quad \forall S \subseteq N. \end{aligned}$$

has optimal objective function value $v(N)$.

Indeed, suppose $S_1, S_2, \dots, S_t \subseteq N$ are the coalitions “selected” by an optimal solution of (2.3), i.e., $y_{S_i} = 1$, for $i = 1, 2, \dots, t$ and $y_S = 0$ for $S \neq S_1, S_2, \dots, S_t$. Then $S_i \cap S_j = \emptyset$, for $i \neq j$ (since each item can only be packed into one bin). The optimal objective function value is $\sum_{i=1}^t v(S_i)$. But this must equal $v(N)$, since, by superadditivity,

$$\sum_{i=1}^t v(S_i) \leq v(N).$$

Let us denote by $v'(N)$ the optimal objective function value of (2.1). As explained above, $v(N)$ is the optimal objective function value of its 0-1 integer linear program (2.3). The necessary and sufficient condition for the non-emptiness of the ϵ -core is given below (cf. [28] for the uniform bin packing game). The proof is identical to the one given by Faigle and Kern [28]. We include it for convenience of the reader.

Lemma 2.2. *Assume a game $\langle N, v \rangle$ is superadditive and $v \geq 0$. Given $\epsilon \in [0, 1]$, the ϵ -core of N is nonempty if and only if $\epsilon \geq 1 - v(N)/v'(N)$.*

Proof. (\Rightarrow) Recall that $x \in \mathbb{R}^N$ is in the ϵ -core of N if and only if $x(S) \geq (1 - \epsilon)v(S)$, $\forall S \subseteq N$ and $x(N) = v(N)$. Note that $n = |N|$ and $x(S) = \sum_{i \in S} x_i$. Therefore, if x is in the ϵ -core, then $x/(1 - \epsilon)$ must be a feasible solution to (2.1), implying

$$\frac{v(N)}{1 - \epsilon} = \frac{x(N)}{1 - \epsilon} \geq v'(N),$$

which yields $\epsilon \geq 1 - v(N)/v'(N)$.

(\Leftarrow) Assume $\epsilon \geq 1 - v(N)/v'(N)$ is true. Let $\bar{\epsilon} = 1 - v(N)/v'(N)$, hence $\epsilon \geq \bar{\epsilon}$, and let \bar{x} be an optimal solution of (2.1). We claim $x = (1 - \bar{\epsilon})\bar{x}$ is in the ϵ -core of N , by verifying the two conditions as below:

$$x(S) = (1 - \bar{\epsilon})\bar{x}(S) \geq (1 - \bar{\epsilon})v(S) \geq (1 - \epsilon)v(S), \quad \forall S \subseteq N$$

and

$$x(N) = (1 - \bar{\epsilon})\bar{x}(N) = (1 - \bar{\epsilon})v'(N) = v(N).$$

□

As a direct corollary of Lemma 2.2, the balancedness of a cooperative game can be characterized by the integrality gap of its characteristic function.

Corollary 2.3. *A superadditive game is balanced if and only if $v(N) = v'(N)$.*

We refer to $\epsilon_N := 1 - v(N)/v'(N)$ as the minimal taxation rate of a game $\langle N, v \rangle$. Thus, the minimal taxation rate of the superadditive game with a nonnegative value function is determined by the relative integrality gap. Analyzing the relative integrality gap thus builds a connection between cost allocation problems and fractional programming problems (cf. the last paragraph of Section 2.3).

Analogous results can be derived for subadditive cost games: We define their ϵ -core to be the set of allocation vectors x satisfying

- (i) $x(N) = c(N)$,
- (ii'') $x(S) \leq (1 + \epsilon)c(S)$.

For simplicity, we still call ϵ the tax rate (as the idea behind the ϵ -core is essentially the same for the two cases).

The formulation of the core allocation problem w.r.t. to cost allocation is as follows:

$$\begin{aligned} & \text{maximize } x(N) && (2.4) \\ & \text{subject to } x(S) \leq c(S), \quad \forall S \subseteq N, \\ & \quad \quad \quad x(S) \geq 0, \quad \forall S \subseteq N. \end{aligned}$$

Denote by $c'(N)$ the optimal objective function value of (2.4). Similarly, the non-emptiness of the ϵ -core can be characterized as below.

Lemma 2.4. *Assume the cost game $\langle N, c \rangle$ is subadditive and $c \geq 0$. Given $\epsilon > 0$, the ϵ -core of N is nonempty if and only if $\epsilon \geq c(N)/c'(N) - 1$.*

2.3 OPERATIONS RESEARCH GAMES

Operations research games are games where the value function $v(S)$ equals the optimum value of an optimization problem defined on S , as is the case, *e.g.*, MST game and bin packing game (*cf.* Section 2.1).

It is interesting to study cooperative games from an algorithmic view: For any game, we compute the solution concepts arised from cooperative game theory, *e.g.*, core, nucleolus, Shapley value etc. For some games, testing the non-emptiness of the core is NP-complete and even testing the core membership of an allocation is NP-complete. As we have explained before, the ϵ -core can be seen as an ϵ -approximation to the core, the minimal taxation rate therefore provides a best possible approximation ratio to the core.

Exhibiting an allocation x in the ϵ -core, even when ϵ is not as small as possible, is interesting from a game theoretical perspective. Proving the existence of such an ϵ -core allocation requires to show that $v(N) \geq (1 - \epsilon)v'(N)$, *i.e.*, to exhibit a solution y of (2.3) of value at least $(1 - \epsilon)v'(N)$. As $v'(N)$ is an upper bound of (2.3), this means that, in particular, y must be an $(1 - \epsilon)$ approximate solution of (2.3). So our game theoretic approach may also yield new insights into approximation aspects. For instance, the proof of non-emptiness of the $1/4$ -core in Chapter 3 implies an efficient $3/4$ -approximation algorithm for multiple subset sum. In the following, we focus on two specific problems: bin packing games (Chapter 3 and Chapter 4) and location planning (Chapter 5).

(Uniform) bin packing game

For many years, logistics and supply chain management are playing an important role in both industry and our daily life. In view of the big profit generated in this area, the question therefore arises how to fairly allocate profits among the players that are involved. Take online shopping as an example: Goods are delivered by means of carriers. Often, shipping costs are proportional to the weight or volume of the goods, and the total cost is basically determined by the competitors. But there might be more subtle ways to compute “fair” shipping cost (and allocation between senders and receivers). It is natural to study such allocation problems in the framework of cooperative games. As a first step in this direction, we analyze a simplified model with uniform packet sizes in this chapter.

We adopt the concept of the ϵ -core and try to find the minimal ϵ that guarantees a nonempty ϵ -core for all instances of the bin packing game. By Lemma 2.2, finding such a minimal ϵ is indeed equivalent to finding a best lower bound on the relative integrality gap, i.e., $v(N)/v'(N)$. In this chapter, we analyze the relative integrality gap for the uniform bin packing game, where the main results obtained here are based on papers [KQ12a] and [KQ13a].

In Section 3.1, we formally describe the definition of the uniform bin packing game and introduce the concept of fractional packing. After that, we discuss three LP rounding approaches – greedy selection, modified greedy selection and greedy packing in Sections 3.2, 3.3, 3.4 respectively. Finally, in Section 3.6, we combine the greedy selection

and greedy packing to prove that the $1/4$ -core is nonempty for any instance of the bin packing game.

3.1 PROBLEM FORMULATION

As motivated at the beginning of this chapter, we study a specific game of the following kind: There are two disjoint sets of players, say, A and B . Each player $i \in A$ possesses an item of value/size a_i , for $i = 1, \dots, n$, and each player $j \in B$ possesses a truck/bin of capacity 1. The items produce a profit proportional to their size a_i if they are brought to the market place. The value $v(N)$ of the grand coalition $N = A \cup B$ thus represents the maximum profit achievable. The question now is to determine how $v(N)$ should be allocated to the owners of the items and the owners of the trucks.

Previous results on the uniform bin packing game can be summarized as follows. Woeginger [74] showed that the $1/3$ -core of the bin packing game is nonempty. Faigle and Kern [28] showed that the integrality gap, defined by the difference of the optimum fractional packing value and the optimum integral packing value (*cf.* Section 3.1.1) is bounded by $1/4$, if all item sizes are strictly larger than $1/3$, thereby implying that the $1/7$ -core is nonempty in that case (which was independently shown by Kuipers [50]). Moreover, in the general case, given a fixed $\epsilon \in (0, 1)$, they proved that the ϵ -core is always non-empty if the number of bins is sufficiently large ($\geq O(\epsilon^{-5})$). Liu [54] presents complexity results on testing emptiness of the core and core membership for the bin packing game, stating that both problems are NP-complete. Moreover, the problem of approximating the maximum packing value $v(N)$ is also studied in the literature (called "multiple subset sum problem"): A polynomial time approximation scheme (PTAS) and a $3/4$ approximation algorithm are proposed in [10] and [11] respectively. Other variants of the bin packing game can also be found in [3, 75, 26] etc.

3.1.1 Integral and fractional packing

Formally, a *bin packing game* is defined by a set A of n items $1, 2, \dots, n$ of sizes a_1, a_2, \dots, a_n , and a set B of k bins, of capacity 1 each, where we assume, w.l.o.g., $0 \leq a_i \leq 1$.

A *feasible packing* of an item set $A' \subseteq A$ into a set of bins $B' \subseteq B$ is an assignment of some (or all) elements in A' to the bins in B' such that the total size of items assigned to any bin does not exceed its capacity. Items that are assigned to a bin are called *packed* and items that are not assigned are called *not packed*. The *value* or *size* of a feasible packing is the total size of packed items.

The player set N consists of all items and all bins. The value $v(S)$ of a coalition $S \subseteq N$, where $S = A_S \cup B_S$ with $A_S \subseteq A$ and $B_S \subseteq B$, is the maximum value of all feasible packings of A_S into B_S . A corresponding packing with maximum value is called an *optimum packing*.

Let F be an item set, and denote by $a_F = \sum_{i \in F} a_i$ the *value* of F . F is called a *feasible set* if $a_F \leq 1$. Denote by \mathcal{F} the set of all feasible sets, w.r.t. all items of N . Let $y_F \in \{0, 1\}$ indicate whether a feasible set F is packed. Then the total earning $v(N)$ of the grand coalition N can be formulated as the following integer linear program:

$$\begin{aligned}
 & \text{maximize} && \sum_{F \in \mathcal{F}} a_F y_F && (3.1) \\
 & \text{subject to} && \sum_{F \ni i} y_F \leq 1, && i = 1, 2, \dots, n, \\
 & && \sum_{F \in \mathcal{F}} y_F \leq k, \\
 & && y_F \in \{0, 1\}, && \forall F \in \mathcal{F}.
 \end{aligned}$$

The first set of constraints ensures that each item is packed to at most one bin and the second constraint guarantees that the total number of bins filled by items is at most k .

Readers may have noticed that problem (3.1) coincides with the ILP formulation of the multiple subset sum problem (*cf.* Chapter 1), which is NP-hard. The bin packing game is actually the game theoretical

model of multiple subset sum, whose main interest is to find fair allocations for (players representing) items and bins, instead of finding the maximum packing value. For this purpose, we first consider the LP-relaxation of (3.1):

$$\begin{aligned}
 & \text{maximize } \sum_{F \in \mathcal{F}} a_F y_F & (3.2) \\
 & \text{subject to } \sum_{F \ni i} y_F \leq 1, \quad i = 1, 2, \dots, n, \\
 & \sum_{F \in \mathcal{F}} y_F \leq k, \\
 & y_F \geq 0, \quad \forall F \in \mathcal{F}.
 \end{aligned}$$

A *fractional packing* is a vector $y = (y_F)$ satisfying all constraints of the linear program (3.2). We call a feasible set F *selected/packed* by a packing $y' = (y'_F)$ if $y'_F > 0$. Accordingly, we refer to the “feasible packing” of (3.1) as *integral packing*.

3.1.2 Non-emptiness of the ϵ -core

We learned in Chapter 2 that the ϵ -core is nonempty if and only if $\epsilon \geq 1 - v(N)/v'(N)$, where $v(N)$ is the total gain of the grand coalition N and $v'(N)$ is the corresponding fractional optimum defined by (2.2) (or equivalently the optimum value of the core allocation problem (2.1)). For the bin packing game N , now $v'(N)$ is indeed the value of an optimum fractional packing. Thus, to analyze the minimal taxation rate is indeed to analyze the (relative) gap between the optimum integral packing and the optimum fractional packing. In the following, we show the equivalence of (2.1) and (3.2).

Recall the core allocation problem in Chapter 2 as below:

$$\begin{aligned}
 & \text{minimize } x(N) & (3.3) \\
 & \text{subject to } x(S) \geq v(S), \quad \forall S \subseteq N.
 \end{aligned}$$

By symmetry, we assume w.l.o.g. that there exists an optimal solution allocating the same amount to each bin. Furthermore, it apparently

suffices to consider only those restrictions $x(S) \geq v(S)$ where S consists of exactly one bin and some feasible set. Let x_i be the payoff of item i for $i = 1, 2, \dots, n$ and x_0 be the payoff of each bin. Now the core allocation problem (3.3) can be written in the form

$$\begin{aligned} & \text{minimize } kx_0 + \sum_{i=1}^n x_i && (3.4) \\ & \text{subject to } x_0 + \sum_{i \in F} x_i \geq \alpha_F, \quad \forall F \in \mathcal{F}, \\ & x_i \geq 0, \quad \forall i = 0, 1, \dots, n, \end{aligned}$$

which is the dual problem of (3.2).

We consider following example as an illustration of the fractional packing.

Example 3.1. *Given two bins and four items of sizes $\alpha_i = 1/2$ for $i = 1, 2, 3$ and $\alpha_4 = 1/2 + \epsilon$, with a small $\epsilon > 0$. The optimum fractional packing is described as Figure 3.2.*

Obviously, packing items 1, 2 into the first bin and packing item 4 into the second bin results in an optimum integral packing (see Fig. 3.1) of total value $v(N) = 3/2 + \epsilon$. Let $F_1 = \{1, 2\}$, $F_2 = \{2, 3\}$, $F_3 = \{1, 3\}$, $F_4 = \{4\}$. By solving the linear program (3.2), the optimum fractional packing (Figure 3.2) $y' = (y'_F)$ selects F_1, \dots, F_4 with a fraction $1/2$ each, resulting in a value

$$v'(N) = \sum_{j=1}^4 y'_{F_j} \alpha_{F_j} = \frac{7}{4} + \frac{\epsilon}{2} > v(N).$$

In this example, the minimal taxation rate is

$$\epsilon_N := 1 - \frac{v(N)}{v'(N)} = \frac{1 - 3\epsilon}{7 + \epsilon} < \frac{1}{7},$$

implying that the $1/7$ -core is nonempty.

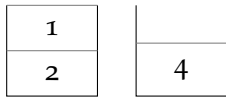


Figure 3.1: Integral opt.

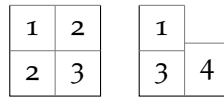


Figure 3.2: Fractional opt.

3.1.3 Reducing the problem size

Trivially, if all items are packed in a feasible integral packing, we get $v(N) = v'(N)$, implying that $\epsilon_N = 0$, so the core is nonempty. Thus let us assume in what follows that no feasible integral packing packs all items. Clearly, any feasible integral packing y with corresponding packed sets F_1, \dots, F_k yields a lower bound $v(N) \geq w(y) := \sum_{i=1}^k \alpha_{F_i}$, where $w(y)$ denotes the total size of the packing y . In view of Lemma 2.2 we are particularly interested in integral packings y of value $w(y) \geq (1 - \epsilon)v'(N)$ for certain $\epsilon > 0$.

For $\epsilon = 1/2$, such integral packings are easy to find by means of a *simple greedy packing* heuristic, that constructs a feasible set F_j to be packed into bin j for $j = 1, 2, \dots, k$ in the following way:

SIMPLE PACKING

Input: bin j , items a_{i_1}, \dots, a_{i_t} .

If $\sum_{s=1}^t a_{i_s} \leq 1$ Then return $\{a_{i_1}, \dots, a_{i_t}\}$;

Else

Let $a_{i_1} + \dots + a_{i_r} \leq 1, a_{i_1} + \dots + a_{i_{r+1}} > 1$;

Return the larger of $\{a_{i_1}, \dots, a_{i_r}\}$ and $\{a_{i_{r+1}}\}$;

End

First order the available (yet unpacked) items non-increasingly, say, $a_{i_1} \geq a_{i_2} \geq \dots \geq a_{i_t}$. Then, starting with $F_j = \emptyset$, keep adding items from the list using Simple Packing. Clearly, this eventually yields a feasible set F_j of size $> 1/2$.

Indeed – unless all items get packed (which we assume to be impossible) – the final F_j has size $> 1 - a$, where a is the minimum size of unpacked items. Applying greedy packing to all bins will exhibit an integral packing y with $\alpha_{F_j} > 1/2$ for all j , so $w(y) > k/2 \geq v'(N)/2$,

thus proving non-emptiness of the $1/2$ -core by Lemma 2.2. A bit more work is required to exhibit an integral packing y with $w(y) \geq \frac{2}{3}v'(N)$ (cf. Section 3.2).

Denote by $\epsilon_N = 1 - v(N)/v'(N)$ the minimal taxation rate of a bin packing game N . For convenience of description, we often abbreviate $\langle N, v \rangle$ to N and refer to N as an instance of the bin packing game. We thus seek for good lower bounds on $v(N)/v'(N)$. The first step in [74] is to reduce the analysis to item sizes $a_i > 1/3$ for all i . Similarly, if we aim for a bound $\epsilon_N \leq \epsilon$ with $\epsilon \in [1/4, 1/3)$, it suffices to investigate instances with item sizes $a_i > 1/4$, as can be seen from the following two lemmas.

Lemma 3.2. *Let S^+ be a set of items disjoint from N and let $a_{S^+} = \sum_{i \in S^+} a_i$ be the total size of S^+ . Then $v(N) + a_{S^+} = v(N \cup S^+)$ implies $\epsilon_{N \cup S^+} \leq \epsilon_N$.*

Proof. From Lemma 2.2 we know $\epsilon_N = 1 - v(N)/v'(N)$. Thus, $v(N) + a_{S^+} = v(N \cup S^+)$ implies

$$\epsilon_{N \cup S^+} = 1 - \frac{v(N \cup S^+)}{v'(N \cup S^+)} \leq 1 - \frac{v(N) + a_{S^+}}{v'(N) + a_{S^+}} \leq 1 - \frac{v(N)}{v'(N)} = \epsilon_N.$$

□

Lemma 3.3. *Let N be a bin packing game and assume that N is ϵ -balanced for some $\epsilon < 1/2$. Then adding “small” items of size $a_i \leq \epsilon$ does not affect ϵ -balancedness.*

Proof. First note that it suffices to prove the claim in case where a single small item i_0 is added. Let $N^+ := N \cup \{i_0\}$ denote the extended game. If i_0 can be packed “on top of” the optimum integral packing for N (i.e. some bin j is filled only up to at most $1 - a_{i_0}$). In this case, we conclude that $v(N^+) = v(N) + a_{i_0}$. By Lemma 3.2, the claim is true. Otherwise, if i_0 cannot be packed on top of the optimum integral packing for N , then each bin must be filled to at least $1 - a_{i_0} \geq 1 - \epsilon$. Hence, $v(N^+) \geq (1 - a_{i_0})k \geq (1 - \epsilon)k \geq (1 - \epsilon)v'(N^+)$, implying that the ϵ -core is nonempty. □

Thus in what follows, when seeking for an upper bound $\epsilon_N \leq \epsilon$ with $\epsilon \in [1/4, 1/3)$, we may assume that all item sizes are at least $\alpha_i > 1/4$. (This is actually a rather interesting class anyway, as it contains all instances of 3-Partition, cf. Section 3.7).

3.2 GREEDY SELECTION

We first present an alternative proof for the fact that the $1/3$ -core of the bin packing game is nonempty, using an LP rounding approach that we call *greedy selection*. Consider any instance N of the bin packing game, which consists of k bins and n items of sizes $\alpha_1, \dots, \alpha_n$ with all $\alpha_i > 1/4$. Although, for the purpose of this section, it would suffice to assume $\alpha_i > 1/3$ and the corresponding proof will be much easier. The reason we consider $\alpha_i > 1/4$ is to further explain that this greedy selection approach is not able to improve the $1/3$ -bound, however, an improved version will be presented in Section 3.3. Note that $\alpha_i > 1/4$ implies that any feasible set contains at most 3 items.

3.2.1 The basic idea

Let $y = (y_F)_{F \in \mathcal{F}}$ be an optimal fractional packing. Let \mathcal{F} be the support of y , i.e., $\mathcal{F} := \text{supp } y = \{F \mid y_F > 0\}$. First note that if $\alpha_F \leq 2/3$ for all $F \in \mathcal{F}$ then $v'(N) \leq \frac{2}{3}k$, and hence any integral packing filling each bin to at least $1/2$ would achieve a total value $\geq k/2$. Thus, $v(N) \geq \frac{k}{2} \geq \frac{3}{4}v'(N)$, proving non-emptiness even for the $1/4$ -core. More generally, as we will see below, to extract a reasonably good integral packing from the fractional packing y , we may focus on $\bar{\mathcal{F}} := \{F \in \mathcal{F} \mid \alpha_F > \frac{2}{3}\}$, the “interesting part” of the support of y . So assume $\bar{\mathcal{F}} \neq \emptyset$, i.e., it has nonzero length $l := \sum_{F \in \bar{\mathcal{F}}} y_F$. Let $\bar{\mathcal{F}} = \{F_1, \dots, F_m\}$ and

$$\alpha_{F_1} \geq \alpha_{F_2} \geq \dots \geq \alpha_{F_m} > \frac{2}{3}.$$

Note that the number of fully packed items is at most $3k$ (3 items per bin), so that $m = |\text{supp } y| \leq 3k + 1$ for a basic feasible solution y of (3.2), where $\text{supp } y := \{F \mid y_F > 0\}$.

The basic idea is to construct an integral solution “greedily”, i.e., starting with F_1 , we construct a sequence of feasible sets by choosing in each step the largest size F_i in $\bar{\mathcal{F}}$ that is disjoint from all previously chosen ones. Formally: Start with $s = 1$ and do the following while $\bar{\mathcal{F}} \neq \emptyset$: Let F_{i_s} be the largest size set in $\bar{\mathcal{F}}$ and $\bar{\mathcal{F}}_{i_s} := \{F \in \bar{\mathcal{F}} \mid F \cap F_{i_s} = \emptyset\}$. Replace $\bar{\mathcal{F}}$ by $\bar{\mathcal{F}} \setminus \bar{\mathcal{F}}_{i_s}$ and s by $s + 1$. Let F_{i_1}, \dots, F_{i_r} denote the sequence constructed in this way. To illustrate this, we consider an example to see how this works.

Example 3.4. *Given a bin packing game instance of three bins and a fractional packing (Figure 3.3), where feasible sets F_1, F_2, \dots are ordered non-increasingly (according to their sizes). (* denotes unspecified items.) The integral packing obtained by greedy selection is as described in Figure 3.4.*

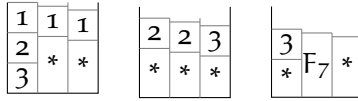


Figure 3.3: Fractional packing.

The greedy selection starts with selecting $F_{i_1} := F_1$, then find the next feasible set $F_{i_2} := F_7$, which is disjoint from F_1 . Thus, we can pack F_{i_1}, F_{i_2} to two bins and for the last bin, we can “greedily” fill it (in an obvious way, cf. Section 3.1.3) to at least half its capacity, resulting in a feasible integral packing as below (Figure 3.4)

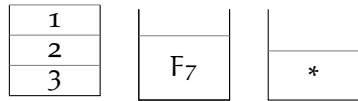


Figure 3.4: Integral packing obtained by greedy selection.

We can roughly estimate the relative integrality gap (of this example) as follows: $v'(N)$ is bounded by $3a_{F_1}$, and the integral packing obtained by greedy selection has value at least

$$a_{F_1} + a_{F_7} + \frac{1}{2} \geq a_{F_1} + 1 \geq 2a_{F_1} \geq \frac{2}{3}v'(N),$$

implying that the $1/3$ -core is nonempty.

3.2.2 Proof of $1/3$ -core $\neq \emptyset$

Now we show that the previous greedy selection yields an integral packing of total value at least $\frac{2}{3}v'(N)$. Define the *length* of \tilde{F}_{i_s} to be $l_{i_s} := \sum_{F \in \tilde{F}_{i_s}} y_F$ and the *value* to be $v_{i_s} := \sum_{F \in \tilde{F}_{i_s}} y_F a_F$. As each F_{i_s} contains at most 3 items, say, $F_{i_s} = \{j_1, j_2, j_3\}$, we find that

$$\begin{aligned} l_{i_s} &= \sum_{F \in \tilde{F}_{i_s}} y_F \\ &\leq \sum_{F \ni j_1} y_F + \sum_{F \ni j_2} y_F + \sum_{F \ni j_3} y_F - 2y_{F_{i_s}} \\ &\leq 3 - 2y_{F_{i_s}} < 3. \end{aligned} \tag{3.5}$$

Note that when F_{i_s} contains less, say, only two items, the same inequality $\sum_{F \in \tilde{F}_{i_s}} y_F \leq 2 - y_{F_{i_s}} \leq 3 - 2y_{F_{i_s}}$ holds.

Hence in each step, when removing \tilde{F}_{i_s} , we remove at most 3 from the total length $l = \sum_{F \in \mathcal{F}} y_F$, so that our construction yields F_{i_1}, \dots, F_{i_r} with $r \geq l/3$. By the greedy choice of F_{i_s} we have $l_{i_s} a_{F_{i_s}} \geq v_{i_s}$. Hence

$$a_{F_{i_s}} = \frac{l_{i_s}}{3} a_{F_{i_s}} + (1 - \frac{l_{i_s}}{3}) a_{F_{i_s}} \geq \frac{1}{3} v_{i_s} + (1 - \frac{l_{i_s}}{3}) \frac{2}{3}.$$

Summation yields

$$a_{F_{i_1}} + \dots + a_{F_{i_r}} \geq \frac{1}{3} \sum_{F \in \mathcal{F}} y_F a_F + (r - \frac{l}{3}) \frac{2}{3}.$$

We extend this greedy selection by $(k - r)$ bins, each filled to at least $1/2$. As $r \geq \frac{l}{3}$, the resulting packing implies

$$\begin{aligned}
 v(N) &\geq \frac{1}{3} \sum_{F \in \mathcal{F}} y_F a_F + (r - \frac{l}{3}) \frac{2}{3} + (k - r) \frac{1}{2} \\
 &\geq \frac{1}{3} \sum_{F \in \mathcal{F}} y_F a_F + (k - \frac{l}{3}) \frac{1}{2} \\
 &= \frac{1}{3} \sum_{F \in \mathcal{F}} y_F a_F + \frac{1}{3} l + \frac{1}{2} (k - l) \\
 &\geq \frac{2}{3} \sum_{F \in \mathcal{F}} y_F a_F + \frac{1}{2} (k - l), \tag{3.6}
 \end{aligned}$$

whereas

$$v'(N) \leq \sum_{F \in \mathcal{F}} y_F a_F + \frac{2}{3} (k - l).$$

Hence $v(N)/v'(N) \geq 2/3$, as claimed.

We also learn from the above proof that even if we consider the instance with all item sizes strictly larger than $1/4$, the greedy selection only yields the non-emptiness of the $1/3$ -core (based on the current analysis). However, with a little bit more effort, the greedy selection approach is capable of improving the current result. Details be discussed in the next section – the modified greedy selection.

3.3 THE MODIFIED GREEDY SELECTION

Matsui [58] claimed that the bound $1/3$ for the minimal taxation rate is tight, *i.e.*, there exists an instance of the bin packing game such that the ϵ -core is empty for any $\epsilon < 1/3$. In his proof, a bin packing game instance G_α of 3 bins and 5 items of sizes $1/2, 1/2, 1/2, 1/2 + \alpha, 1/2 + \alpha$ ($0 \leq \alpha \leq 1/2$) is considered. He “showed” that for any given $\epsilon < 1/3$, by properly choosing α , the ϵ -core of G_α is always empty, based on the fact that items 1, 2, 3 (with size $1/2$ each) cannot all be packed in an optimum integral packing. To this end he claims that an ϵ -core allocation must allocate 0 for each of the 3 players corresponding to items of size $1/2$. This implication is only true when one seeks for a

core allocation (with $\epsilon = 0$) while obviously incorrect in case of the ϵ -core allocation.

In this section, we aim to improve the bound $\epsilon_{\min} \leq 1/3$ by rounding the fractional packing w.r.t. a modified ordering of its selected feasible sets. First recall that for F_{i_s} , we have the inequality (3.5)

$$l_{i_s} = \sum_{F \in \bar{\mathcal{F}}_{i_s}} y_F \leq 3 - 2y_{F_{i_s}}.$$

Summation thus yields

$$l = \sum_{s=1}^r l_{i_s} \leq \sum_{s=1}^r (3 - 2y_{F_{i_s}}) = 3r - 2 \sum_{s=1}^r y_{F_{i_s}}.$$

Thus, if $\alpha = \sum_{s=1}^r y_{F_{i_s}}$, we find

$$r \geq \frac{1}{3}(l + 2\alpha). \quad (3.7)$$

The estimate in Section 3.2 can be (slightly) improved by modifying the greedy selection so as to give higher priority to feasible sets $F \in \bar{\mathcal{F}}$ with comparatively large y_F – and thus hopefully increasing α . To this end we modify the size of each $F \in \bar{\mathcal{F}}$ to $\tilde{a}_F := a_F + \frac{1}{9}y_F \geq a_F$. The sizes of $F \in \mathcal{F} \setminus \bar{\mathcal{F}}$ remain unchanged. We then apply greedy selection to $\bar{\mathcal{F}}$ (ordered according to the modified sizes) to obtain certain $F_{i_1}, \dots, F_{i_r} \in \bar{\mathcal{F}}$ and append $(k - r)$ bins filled to at least $1/2$ as before.

Now let us analyze the greedy selection w.r.t. the modified ordering. Estimating the value \tilde{v} (w.r.t. the modified sizes) of the resulting integral packing as we did in Section 3.2 (now using $r \geq \frac{1}{3} + \frac{2}{3}\alpha$ instead of $r \geq 1/3$), yields

$$\begin{aligned} \tilde{v} &\geq \tilde{a}_{F_{i_1}} + \dots + \tilde{a}_{F_{i_r}} + \frac{1}{2}(k - r) \\ &\geq \frac{1}{3} \sum_{\bar{\mathcal{F}}} y_F \tilde{a}_F + \left(r - \frac{l}{3}\right) \frac{2}{3} + (k - r) \frac{1}{2} \\ &\geq \frac{1}{3} \sum_{\bar{\mathcal{F}}} y_F \tilde{a}_F + \left(\frac{2}{3}\alpha\right) \frac{2}{3} + \left(k - \frac{l}{3} - \frac{2}{3}\alpha\right) \frac{1}{2} \end{aligned}$$

$$\begin{aligned}
 &= \frac{1}{3} \sum_{\bar{\mathcal{F}}} y_F \bar{a}_F + \frac{1}{9} \alpha + (k - \frac{l}{3}) \frac{1}{2} \\
 &= \frac{1}{3} \left(\sum_{\bar{\mathcal{F}}} y_F a_F + \frac{1}{9} \sum_{\bar{\mathcal{F}}} y_F^2 \right) + \frac{1}{9} \alpha + \frac{l}{3} + (k - l) \frac{1}{2} \\
 &\geq \frac{2}{3} \sum_{\bar{\mathcal{F}}} y_F a_F + \frac{1}{27} \sum_{\bar{\mathcal{F}}} y_F^2 + \frac{1}{9} \alpha + (k - l) \frac{1}{2}.
 \end{aligned}$$

By definition of \bar{a} , the true value of our packing equals $\tilde{v} - \frac{1}{9} \alpha$. Subtract $\frac{1}{9} \alpha$ from both sides of the last inequality to conclude

$$v(N) \geq \tilde{v} - \frac{1}{9} \alpha \geq \frac{2}{3} \sum_{\bar{\mathcal{F}}} y_F a_F + \frac{1}{27} \sum_{\bar{\mathcal{F}}} y_F^2 + (k - l) \frac{1}{2}. \quad (3.8)$$

Now write $l = \beta k$ with $\beta \in (0, 1]$. Again, due to the fact that y is assumed to be an optimal basic solution of (3.2), the number of feasible sets $F \in \bar{\mathcal{F}}$ is bounded by $m \leq 3k + 1 \leq 4k$. Hence $\sum_F y_F^2$ can be bounded from below by assuming that all y_F have size $\frac{1}{4k} = \frac{\beta}{4}$ (and their number is $4k$), i.e., $\sum y_F^2 \geq \frac{\beta}{4} l$. Hence (3.8) yields

$$\begin{aligned}
 v(N) &\geq \frac{2}{3} \sum_{\bar{\mathcal{F}}} y_F a_F + \frac{\beta}{108} l + (1 - \beta) \frac{k}{2} \\
 &\geq \left(\frac{2}{3} + \frac{\beta}{108} \right) \sum_{\bar{\mathcal{F}}} y_F a_F + (1 - \beta) \frac{k}{2},
 \end{aligned} \quad (3.9)$$

whereas

$$v'(N) \leq \sum_{\bar{\mathcal{F}}} y_F a_F + (k - l) \frac{2}{3} = \sum_{\bar{\mathcal{F}}} y_F a_F + (1 - \beta) \frac{2}{3} k. \quad (3.10)$$

Let $p(\beta)$ and $q(\beta)$ denote the right hand sides in (3.9) and (3.10) respectively, so that $\frac{v(N)}{v'(N)} \geq \frac{p(\beta)}{q(\beta)}$. The worst case occurs when $\beta = 1$ (as p/q is easily seen to be decreasing), resulting in a slightly improved bound $\epsilon \leq 1/3 - 1/108 = 35/108$ for the minimum taxation rate.

Remark. The factor $1/9$ is due to the following: Any increase $\Delta \alpha$ in $\alpha = y_{F_{i_1}} + \dots + y_{F_{i_r}}$ results in an increase of $\Delta r = \frac{2}{3} \Delta \alpha$ in the lower bound for r (cf. (3.7)). This in turn raises the lower bound for

$v(N)$ by $\Delta r(2/3 - 1/2)$ (cf. (3.6)). Thus any increase $\Delta\alpha$ in the total α -value of the selected F_{i_1}, \dots, F_{i_r} yields a *gain* (i.e., increase in the lower bound for v) of $\Delta\alpha/9$. It can be shown that the factor $1/9$ in the definition of modified sizes is optimal in the sense that any alternative choice would lead to a weaker result in our analysis. Yet our analysis is obviously not tight and we expect the true ratio $v(N)/v'(N)$ achieved by the (modified) greedy approach to be significantly better - though very difficult to analyze.

3.4 LARGE ITEMS: $\alpha_i > 1/3$

In this section, we introduce a new rounding approach due to Faigle and Kern [28], which reduces the problem to a maximum weight matching problem if item sizes are strictly larger than $1/3$. A feasible integral packing is obtained via the optimum fractional matching (with cardinality k). We will derive a tight bound on the integrality gap for such instances ($\alpha_i > 1/3$), thereby improving the result of Kuipers [50] (saying that $1/7$ -core $\neq \emptyset$ in this case).

3.4.1 Fractional matching

Since $\alpha_i > 1/3$, at most two items fit into a bin. Hence a fractional packing is close to a fractional matching of items and can thus be treated with well-known techniques from matching theory (cf. Lovász and Plummer [55]). Here we use this idea to analyze the relative integrality gap for this specific case.

Formally, given an instance N of the bin packing game consisting of n items of sizes $\alpha_i > 1/3$, for $i = 1, 2, \dots, n$ and k bins. We construct its *support graph* $G = (V, E)$ of $2n$ nodes as below:

- (a) Nodes $1, 2, \dots, n$ represent n items and each has a value α_i .
Nodes $n + 1, n + 2, \dots, 2n$ represent dummy items of value 0 ;

- (b) For all $i, j \in \{1, 2, \dots, n\}$, $i \neq j$, we add an edge $(i, j) \in E$ if $\alpha_i + \alpha_j \leq 1$ and assign a weight $w_{(i,j)} = \alpha_i + \alpha_j$;
- (c) For all $i \in \{1, 2, \dots, n\}$, we add edges $(i, n+i)$ to E and assign a weight $w_{(i,n+i)} = \alpha_i$.

All edges constructed in (b), (c) correspond to all 2-element and 1-element feasible sets respectively. Thus, the edge set E represents all feasible sets of N . The maximum weight matching with cardinality k therefore represents the maximum (optimum) integral packing (cf. Example 3.5 below).

Example 3.5. Recall Example 3.1 where we are given two bins and four items of sizes $1/2, 1/2, 1/2, 1/2 + \epsilon$, where ϵ is a small positive number. The corresponding maximum weight matching problem is described in Figure 3.5.

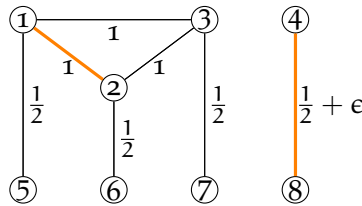


Figure 3.5: A maximum weight matching with cardinality 2.

For all $e \in E$, let w_e be the edge weight and let $y_e \in \{0, 1\}$ indicate whether an edge is in the matching. Thus, the (restricted cardinality) maximum weight matching problem can be formulated as follows:

$$\begin{aligned}
 &\text{maximize } \sum_{e \in E} w_e y_e && (3.11) \\
 &\text{subject to } \sum_{e \ni i} y_e \leq 1, \quad \forall v \in V, \\
 &\quad \quad \quad \sum_{e \in E} y_e \leq k, \\
 &\quad \quad \quad y_e \in \{0, 1\}, \quad \forall e \in E.
 \end{aligned}$$

Similar as before, the corresponding relaxation is

$$\begin{aligned}
 & \text{maximize} && \sum_{e \in E} w_e y_e && (3.12) \\
 & \text{subject to} && \sum_{e \ni i} y_e \leq 1, && \forall v \in V, \\
 & && \sum_{e \in E} y_e \leq k, \\
 & && y_e \geq 0, && \forall e \in E
 \end{aligned}$$

and a feasible solution of (3.12) is called a feasible *fractional matching*.

Thus, integral and fractional matchings correspond to integral and fractional packings respectively. The reason we deal with feasible packing by means of matching is to use the *half-integrality* of fractional matching (cf. Section 3.4.2).

3.4.2 Half-integrality

Maximum weight matching can be solved efficiently and there are nice algorithms for solving this problem, e.g., the blossom algorithm (Jack Edmonds [24]). Here we are interested in the relative integrality gap of the maximum weight matching with cardinality k . First we introduce a well known result from matching theory (cf. Lovász and Plummer [55]).

Lemma 3.6 ([55]). *The optimal fractional matching is half integral, i.e., the basic optimal solution of (3.12) takes values in $\{0, 1/2, 1\}$.*

As before, we denote by $v(N)$, $v'(N)$ the values of the maximum integral (3.11) and fractional matching (3.12) respectively. The idea is that we first consider an optimal fractional matching, then we round it to an integral solution, which is feasible to (3.11) and compare the obtained value with $v'(N)$.

Given an instance N of the bin packing game with item sizes $a_i > 1/3$, let $G = (V, E)$ be the support graph of N (obtained by the reduction (a) - (b) - (c)). Let $y = (y_e) \in \{0, 1/2, 1\}^E$ be an optimal

solution of (3.12). It suffices to consider the fractional components $\mathcal{C} = \{e \in E \mid y_e = 1/2\}$, consisting of cycles and paths.

Lemma 3.7 ([28]). *\mathcal{C} does not contain even paths or odd paths of length strictly larger than one or even cycles.*

Proof. Assume $P \in \mathcal{C}$ is an even path (or an even cycle), consisting of edges e_1, e_2, \dots, e_{2p} . We may set the fraction of all even edges or all odd edges to 1 to obtain a feasible solution with total value no less than before (cf. Figure 3.6). Indeed, we set $y_{e_i} = 1$ for odd numbers $i = 1, 3, \dots, 2p - 1$ if

$$w_{e_1} + w_{e_3} + \dots + w_{e_{2p-1}} \geq w_{e_2} + w_{e_4} + \dots + w_{e_{2p}}.$$

Otherwise we set $y_{e_i} = 1$ for even numbers $i = 2, 4, \dots, 2p$.

If P is an odd path of length $2p + 1 > 1$, we modify $2p$ consecutive edges of P as above, which results in a feasible solution of value no less than before (cf. Figure 3.7). It may happen that the fraction of the remaining edge, say $(2p, 2p + 1)$ must be set to 0 since the total fraction of node $2p$ is 1 due to the previous modification. We thus set the fraction of $(2p + 1, 0)$ to $1/2$, where 0 is the node representing a dummy item (cf. Figure 3.8). Thus, the claim still holds. \square

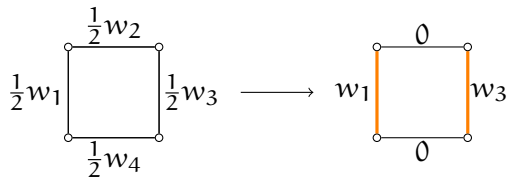


Figure 3.6: Round even cycles: $w_1 + w_3 \geq w_2 + w_4$.

Recall the optimal fractional packing in Example 3.1 (see Figure 3.2): Its graphic representation can be seen from Figure 3.5, where edges $(1, 2)$, $(2, 3)$, $(3, 1)$ and $(4, 8)$ have a fraction $1/2$ each. Thus, this fractional matching consists of one odd cycle and a single edge. Combining above propositions, we obtain the following result stating that non-integral fractional solutions have the same structure as that of Example 3.1.

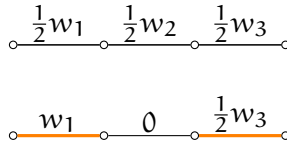


Figure 3.7: Round odd paths: $w_1 \geq w_2$.

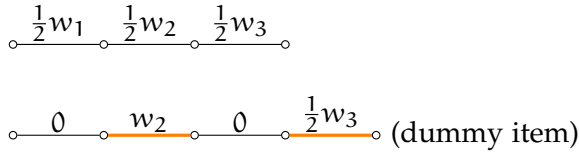


Figure 3.8: Round odd paths: $w_1 < w_2$.

Lemma 3.8 ([28]). \mathcal{C} contains at most one odd cycle and at most one odd path.

Proof. It is trivial to see that if \mathcal{C} contains more than one single edge, say, w.l.o.g., two single edges, we can round the solution by setting the fraction of the edge with the larger weight to 1, and by setting the fraction of the other edge to 0. Now assume \mathcal{C} contains two odd cycles C_1, C_2 and $i_1 \in C_1, i_2 \in C_2$ are nodes of the minimal weight in C_1, C_2 respectively. Thus, $a_{i_1} \leq 1/2, a_{i_2} \leq 1/2$, implying that there is an edge between i_1 and i_2 .

Let $e' = (i_1, i_2)$ and consider the graph G' induced by C_1, C_2 and e' . Denote by l_1, l_2 the length of C_1, C_2 . It can be easily seen that G' has a perfect matching M of cardinality $(l_1 + l_2)/2$, which equals the total fraction of edges in C_1, C_2 , i.e., $\sum_{e \in C_1, C_2} y_e = (l_1 + l_2)/2$. Thus, we can construct a new integral solution by setting the fraction of matched edges by 1 and 0 otherwise. Clearly, the new solution has the same total weight $\sum_{e \in M} w_e = \frac{1}{2} \sum_{e \in C_1, C_2} w_e$ as before (cf. Figure 3.9). \square

We may also assume w.l.o.g that \mathcal{C} contains an odd cycle and a single edge. Indeed, if \mathcal{C} only contains an odd cycle C (or a single edge), then $\sum_{e \in \mathcal{C}} y_e \leq k - 1/2$. We can select an arbitrary edge e' which is disjoint from C (i.e., $e' \notin C, e'$ and C do not have common

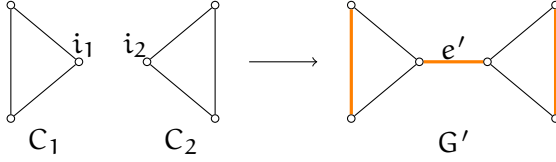


Figure 3.9: Round two odd cycles.

nodes) and let $y_{e'} = 1/2$. As we assumed in Section 3.1.3 that not all items can be packed in an feasible packing, such an e' can always be found.

3.4.3 Tight lower bound

Having achieved the above simplifications, we are now able to analyze the relative relative gap. The following result improves the result of Kuipers [50] (stating that the $1/7$ -core is nonempty in this case).

Theorem 3.9. *If $\alpha_i > 1/3$ for all i , then $\frac{v(N)}{v'(N)} \geq \frac{2k+2}{2k+3}$.*

Proof. By Lemma 3.8, \mathcal{C} contains an odd cycle and a single edge. Let us denote by C the odd cycle and denote by $e' = (i, j)$ the single edge in \mathcal{C} . Let $\mathcal{D} = \{e \in E \mid y_e = 1\}$. We have

$$v'(N) = \frac{1}{2} \sum_{e \in C} w_e + \frac{w_{e'}}{2} + \sum_{e \in \mathcal{D}} w_e. \quad (3.13)$$

Let s be the node of the minimal weight in C . Thus, $\alpha_s \leq 1/2$ and the maximum weight matching M on C does not contain s . We round the fractional components of y by setting $y_e = 1$ for all $e \in M$ and $y_e = 0$ for all $e \in C \setminus M$. Moreover, we let $y_{e'} = 1$. It can be easily seen that the new solution is feasible to (3.11). Hence,

$$v(N) \geq \sum_{e \in M} w_e + \sum_{e \in \mathcal{D}} w_e + w_{e'}. \quad (3.14)$$

Now equation (3.13) can be written as

$$v'(N) = \sum_{e \in M} w_e + \alpha_s + \frac{w_{e'}}{2} + \sum_{e \in \mathcal{D}} w_e. \quad (3.15)$$

Observe that $w_e \geq 2a_s$ for all $e \in \mathcal{C}$ and that $w_e > 1/2$ for $e \in \mathcal{D}$ (again by assumption that $v(\mathbf{N}) \neq v'(\mathbf{N})$, cf. Section 3.1.3). We have

$$\sum_{e \in \mathcal{M}} w_e \geq 2a_s \cdot |\mathcal{M}|, \quad (3.16)$$

$$\sum_{e \in \mathcal{D}} w_e \geq \frac{1}{2} \cdot (k-1-|\mathcal{M}|). \quad (3.17)$$

Let $r := |\mathcal{M}|$. Then $r \geq 1$. Also, we can assume $w_{e'} + a_s > 1$. In fact, if $w_{e'} + a_s \leq 1$, e' has a dummy node (since any 3 items have total size larger than one) and there must be an edge e'' connecting s and the non-dummy node in e' . Thus, setting $y_{e''} = 1$ and $y_{e'} = 0$ yields a matching with a larger total weight, a contradiction. Combining (3.14 – 3.17), we get

$$\begin{aligned} \frac{v(\mathbf{N})}{v'(\mathbf{N})} &\geq \frac{2a_s r + (k-1-r)/2 + w_{e'}}{2a_s r + (k-1-r)/2 + w_{e'}/2 + a_s} \\ &\geq \frac{2a_s + (k-2)/2 + w_{e'}}{2a_s + (k-2)/2 + w_{e'}/2 + a_s} \\ &\geq \frac{2a_s + (k-2)/2 + 1 - a_s}{2a_s + (k-2)/2 + (1-a_s)/2 + a_s} \\ &\geq \frac{2k+2}{2k+3'} \end{aligned}$$

where the last inequality holds since $a_s \leq 1/2$. \square

It follows from Lemma 2.2 that for a fixed k , the $1/(2k+3)$ -core is always nonempty, which is a more general result than that by Kuipers [50], stating that the $1/7$ -core is always nonempty in this case. Moreover, the following example shows that this bound is tight (by letting $\epsilon \rightarrow 0$).

Example 3.10. Consider k bins and $k+2$ items of sizes $a_i = 1/2$ for $i = 1, 2, 3$ and $a_i = 1/2 + \epsilon$ for $i = 4, 5, \dots, k+2$.

Clearly, the optimum integral packing packs items 1, 2 to the first bin and items of size $1/2 + \epsilon$ to remaining bins, which results in a total value

$$v(\mathbf{N}) = 1 + (k-1)\left(\frac{1}{2} + \epsilon\right).$$

Similar to Example 3.1, the optimum fractional packing packs feasible sets $\{1, 2\}$, $\{2, 3\}$, $\{1, 3\}$, $\{4\}$ to the first two bins, with a fraction $1/2$ each and packs items of size $1/2 + \epsilon$ to remaining bins. This yields

$$v'(N) = 1 + \frac{1}{2}(1 + \frac{1}{2} + \epsilon) + (k-2)(\frac{1}{2} + \epsilon).$$

Thus,

$$\frac{v(N)}{v'(N)} = \frac{2k+2 + (4k-4)\epsilon}{2k+3 + (4k-6)\epsilon} \approx \frac{2k+2}{2k+3}.$$

The above proof of the tight bound on the relative integrality gap for the instance with item sizes strictly larger than $1/3$ can be easily turned into an LP rounding algorithm that finds an integral packing pretty close to the fractional optimum. Thus, one naturally wants to use this approach to improve the result on the minimal taxation rate. However, by Lemma 3.3, we need to consider items of size less than or equal to $1/3$. For such instances, a feasible set may contain 3 items, therefore, the reduction (a)-(b)-(c) fails in this case. To remedy this, we may deal with items of size strictly larger than $1/3$ and items of size less than or equal to $1/3$ separately. Details of this approach will be discussed in the section below.

3.5 MATCHING PLUS GREEDY PACKING

In this section, we try to use the matching approach from the previous section to find an integral packing that approximates the fractional optimum $v'(N)$ with a factor $3/4$. As we have motivated at the end of the last section, we will deal with items of size strictly larger than $1/3$ and other items separately. By Lemma 3.3, it suffices to consider instances with items of size strictly larger than $1/4$.

Formally, we call items of size strictly larger than $1/3$ the *large items* and the *small items* otherwise. Since large items can be handled by the matching approach, the idea is to pack large items first (in an optimal way), by Theorem 3.9, we lose a factor at most $1/(2k+3)$. Then add small items greedily on top of those packed large items. Hopefully

this greedy packing does not lose much on the approximation factor. We show that this algorithm finds an integral packing of value at least $\frac{3}{4}v'(N)$, if the number of bins is at least 16.

3.5.1 *The algorithm*

Let G be the support graph induced by large items (by reduction (a)-(b)-(c), cf. Section 3.4.1). Let $M = \{e_1, e_2, \dots, e_k\}$ be the maximum weight matching of cardinality k on G , where $e_i \in M$ corresponds to a feasible set consisting of at most 2 large items. We construct a feasible integral packing by first assigning e_1, e_2, \dots, e_k to k bins, then greedily fill small items “on top of” each bin, *i.e.*, for each bin j and for all small items i , if $w_{e_j} + a_i \leq 1$, we pack i to bin j . There is no need to specify how exactly the greedy packing proceeds, as our estimation does not require any specific information from this procedure. For the practical implementation, we may first sort items non-increasingly according to item sizes, then apply *e.g.* Simple Packing (cf. Section 3.1.3) for all bins.

Now Let S_1, S_2 be the sets of packed and unpacked small items respectively. Denote by F_1, F_2, \dots, F_k the feasible sets packed to k bins and assume, w.l.o.g.,

$$a_{F_1} \geq a_{F_2} \geq \dots \geq a_{F_k}.$$

If $a_{F_k} \geq 3/4$, we are done. Otherwise, for all feasible sets F_j with $a_{F_j} < 3/4$, we replace the items of F_j by the 3 largest items from S_2 until there are at most two small items unpacked (note that any 3 items of S_2 form a feasible set of value strictly larger than $3/4$).

3.5.2 *The analysis*

Assume that, say, the last $(k - r)$ bins are replaced (each by three items from S_2) in this way for some natural number $r \leq k$. If $a_{F_r} \geq 3/4$, we are done. Hence, we assume that $a_{F_r} < 3/4$. Let $a_M, a_{M'}$ be total

weights of M and the optimum fractional matching M' of G respectively. If all (small) items of S_2 are packed, we have

$$v(N) \geq a_M + a_{S_1} + a_{S_2}.$$

Note that $v'(N) \leq a_{M'} + a_{S_1} + a_{S_2}$ and $a_M/a_{M'} \geq (2k+1)/(2k+3)$ (by Theorem 3.9), this implies

$$\frac{v(N)}{v'(N)} \geq \frac{a_M + a_{S_1} + a_{S_2}}{a_{M'} + a_{S_1} + a_{S_2}} \geq \frac{a_M}{a_{M'}} \geq \frac{2k+1}{2k+3} > \frac{3}{4}.$$

Now consider one or two small items of S_2 remain, the reader can verify that the worst case is that two small items are left. In the following, we assume to the contrary that the $1/4$ -core of N is empty, *i.e.* $v(N) < \frac{3}{4}v'(N)$ and to derive some simplifications of the instance.

Proposition 3.11. $r > \frac{3}{4}k$.

Proof. Let a_0 be the minimal item size of S_2 . Then each bin j must be filled to at least $1 - a_0$ for $j = 1, 2, \dots, r$ (otherwise the item of the minimal size a_0 must be packed already to bin j). Denote by $w := \sum_{j=1}^k a_{F_j}$ the total value of this packing. Hence,

$$w \geq r(1 - a_0) + (k - r) \cdot 3a_0.$$

If $r \leq \frac{3}{4}k$, the right hand side of above inequality attains the minimum when a_0 reaches its lower bound $1/4$. Thus, we get

$$v(N) \geq w \geq \frac{3}{4}r - (k - r) \cdot \frac{3}{4} = \frac{3}{4}k,$$

a contradiction to the assumption that the $1/4$ -core is nonempty. \square

Proposition 3.12. $a_M + a_{S_1} > \frac{2}{3}k$.

Proof. Consider the moment before the replacing procedure, when the packing consists of items corresponding to M and items of S_1 . Since no item $i \in S_2$ can be packed on top of F_1, F_2, \dots, F_k , each bin must be filled to at least $2/3$, proving the claim. \square

3.5.3 *The relative integrality gap*

Let $i_1, i_2 \in S_2$ be the two unpacked small items. Denote by a_{S_1}, a_{S_2} the total size of S_1, S_2 respectively. Thus, the obtained integral packing contains r largest feasible sets, say F_1, F_2, \dots, F_r and all small items from $S_2 \setminus \{i_1, i_2\}$. Hence,

$$\begin{aligned} v(N) &\geq \sum_{i=1}^r a_{F_i} + a_{S_2} - a_{i_1} - a_{i_2} \\ &\geq \frac{r}{k}(a_M + a_{S_1}) + a_{S_2} - \frac{2}{3}. \end{aligned} \quad (3.18)$$

The last inequality holds since $a_{i_1}, a_{i_2} \leq 1/3$. First we observe that the total size of (integrally and fractionally) packed large items in an optimum fractional packing of N is at most $a_{M'}$. Now we use the fact that $a_{M'} - a_M \leq 1/4$ (cf. [28]) and get

$$\begin{aligned} v'(N) &\leq a_{M'} + a_{S_1} + a_{S_2} \\ &\leq a_M + a_{S_1} + a_{S_2} + 1/4. \end{aligned} \quad (3.19)$$

It follows from (3.18), (3.19) and Proposition 3.12 that

$$\begin{aligned} v(N) - \frac{3}{4}v'(N) &\geq \left(\frac{r}{k} - \frac{3}{4}\right)(a_M + a_{S_1}) + \frac{a_{S_2}}{4} - \frac{41}{48} \\ &\geq \left(\frac{r}{k} - \frac{3}{4}\right)\frac{2}{3}k + \frac{a_{S_2}}{4} - \frac{41}{48}. \end{aligned}$$

As $|S_2| = 3(k-r) + 2$, $a_{S_2} \geq |S_2|/4 \geq \frac{3}{4}(k-r) + 1/2$. Thus,

$$v(N) - \frac{3}{4}v'(N) \geq \left(\frac{r}{k} - \frac{3}{4}\right)\frac{2}{3}k + \frac{3}{16}(k-r) - \frac{35}{48}.$$

Observe that the value of the right hand side of above inequality decreases as r decreases. By Proposition 3.11, we get

$$v(N) - \frac{3}{4}v'(N) \geq \frac{3}{64}k - \frac{35}{48}.$$

Thus, $k \geq 16$ implies $v(N) \geq \frac{3}{4}v'(N)$.

For $k < 16$, we do not have any clue to approach this $3/4$ approximation. Since the above analysis can not be carried over for instances with small number of bins. Indeed, the integrality gap $v'(N) - v(N)$ is, roughly speaking, bounded by $1/4$ plus the total size of unpacked small items, which is (roughly) at least $\frac{1}{4} + \frac{1}{3} \times 2 = \frac{11}{12}$ in the worst case (recall the assumption that two small items are unpacked). Thus, the relative gap $v(N)/v'(N)$ might become relatively large when k is small. For instance, if $k = 3$, our estimation is as follows

$$\frac{v(N)}{v'(N)} \leq \frac{(3 - 11/12)}{3} = \frac{36}{25} < \frac{3}{4}'$$

which gives a bad lower bound.

Even though the $3/4$ approximation to $v'(N)$ is not verified for all instances, yet it already yields a $3/4$ approximation to the optimum integral packing $v(N)$ (as the integral optimum can be enumerated in polynomial time if $k \leq 15$ and $\alpha_i > 1/4$). Actually, this algorithm is analogous to the one given by Caprara *et al.* [11]. But we derive this independently and in a different perspective (because our aim is to approximate the fractional optimum).

Moreover, Faigle and Kern proved that the ϵ -core is nonempty if $k \geq 48\epsilon^{-5}$. Taking $\epsilon = 1/4$, it gives $k \geq 49152$. Thus, to show the non-emptiness of the $1/4$ -core, we only need to consider the instance with a fixed size (*i.e.*, $k \leq 49151$ and $\alpha_i > 1/4$). So our analysis significantly improves their result to $k \leq 15$ (for $\epsilon = 1/4$).

Still, our journey to the non-emptiness of the $1/4$ -core continues.

3.6 NON-EMPTINESS OF THE 1/4-CORE

In this section, we aim at constructing an integral packing y of value $w(y) \geq \frac{3}{4}v'(N)$, thus proving the claim that the $1/4$ -core is nonempty for all instances of the uniform bin packing game. We know from previous attempts that neither the greedy selection nor the matching approach yield a $3/4$ approximation to $v'(N)$ (in the analysis). Finally, we succeed this by combining ideas from greedy selection in Section

3.2 and greedy packing via matching in Section 3.4, however, with slight variants.

We first mention two trivial cases, namely $k = 1$ and $k = 2$: Indeed, in case $k = 1$, we obviously have $v(N) = v'(N)$, thus $\epsilon_N = 0$ and in case $k = 2$, the bound $\epsilon_N \leq \frac{1}{4}$ can be seen as follows: Let y' be an optimal fractional solution and let F be a largest (most valuable) feasible set in the support of y' . Then the integral packing y that packs F into the first bin and fills the second bin greedily to at least $1/2$ (as explained earlier) is easily seen to yield a value $w(y) \geq \frac{3}{4}w(y') = \frac{3}{4}v'(N)$, so that $\epsilon_N \leq \frac{1}{4}$.

Throughout this section, we assume that N is a smallest counterexample, *i.e.*, a game with $1/4\text{-core}(N) = \emptyset$ and $n + k$, the number of players is as small as possible. By Lemma 3.3, we can assume that all items of N have size strictly larger than $1/4$.

In the following we reconsider (slight variants of) two packing heuristics that have been introduced earlier in [10] and [45] respectively. The first one, which we call “item packing” (*cf.* Section 3.6.1), seeks to first pack large items, then small ones on top of these, without regarding the optimum fractional solution. The second one, which we call “set packing” (*cf.* Sections 3.6.2, 3.6.3), starts out from the optimum fractional solution y' and seeks to extract an integer packing based on the feasible sets that are (fractionally) packed by y' .

Since N is assumed as a counterexample, the idea is to use item packing and greedy selection to simplify matters: If item packing is failed to prove the desired bound, there must be a few unpacked small items (*cf.* Lemma 3.15). Else, if greedy selection fails, then the number of unpacked small items cannot be too many (*cf.* Lemma 3.17). Thus, the worst case is that the unpacked small items are neither too many nor too few. A combination of item packing and greedy selection will work in this worst case.

3.6.1 Item packing

We first deal with item packing. Recall that an item i is called a *large item* if $a_i > 1/3$ and a *small item* otherwise. Let L and S denote the set of large and small items respectively. If no misunderstanding is possible, we will also interpret L as the game N restricted to the large items (and k bins). Note that at most two large items fit into a bin. Thus packing L is basically a matching problem. This is why we can easily solve it to optimality and why the gap is fairly small (cf. Section 3.4.1). More precisely, Theorem 3.2 from [28] can be stated as

Lemma 3.13 ([28]). $\text{gap}(L) = v'(L) - v(L) \leq \frac{1}{4}$.

This motivates the following Item Packing heuristic:

ITEM PACKING

1. Compute an optimum integral packing y for L .
2. Try to add as many small items on top of y as possible.

There is no need to specify how exactly, small items are added in step 2. A straightforward way would be to sort the small items non-increasingly and apply some first or best fit heuristic. Let $\hat{F}_1, \dots, \hat{F}_k$ denote the feasible sets in the integral packing \hat{y} produced by Item Packing, *i.e.*,

Output Item Packing: $\hat{y} \hat{=} (\hat{F}_1, \dots, \hat{F}_k)$.

Note that, due to Lemma 3.13, Item Packing performs quite well w.r.t. the large items. Thus we expect Item Packing to perform rather well in case there are only a few small items or, more generally, if Item Packing leaves only a few small items unpacked. Let $S_1 \subseteq S$ denote the set of small items that get packed in step 2 and let $S_2 := S \setminus S_1$ be the set of unpacked items. We prove the following bounds:

Lemma 3.14. $\alpha_{\hat{F}_j} > \frac{2}{3}$ for all $j = 1, \dots, k$. Hence, $v(N) > \frac{2}{3}k$ and $v'(N) > \frac{8}{9}k$.

Proof. By definition, the first step of Item Packing produces an optimum integral packing of L of value $v(L)$. Thus the final outcome \hat{y} has value $w(\hat{y}) = v(L) + \alpha_{S_1}$. Hence $v(N) \geq v(L) + \alpha_{S_1}$. The fractional value, on the other hand, is clearly bounded by $v'(N) \leq v'(L) + \alpha_S$. Thus in case $S_2 = \emptyset$ (i.e., $S_1 = S$) we find

$$v'(N) - v(N) \leq v'(L) - v(L) \leq 1/4,$$

implying that

$$\epsilon_N = \frac{v'(N) - v(N)}{v'(N)} \leq \frac{1/4}{k/2} \leq \frac{1}{4} \quad \text{for } k \geq 2,$$

contradicting our assumption that $1/4\text{-core}(N) = \emptyset$. (Note that for $k = 1$ we always have $v = v'$, i.e., $\epsilon_N = 0$.)

Thus we conclude that $S_2 \neq \emptyset$. But this means that the packing \hat{y} produced by Item Packing fills each bin to more than $2/3$, i.e., $\alpha_{\hat{f}_j} > 2/3$ for all j (otherwise, if $\alpha_{\hat{f}_j} \leq 2/3$, any item in S could be packed on top of \hat{f}_j). Hence $v(N) \geq w(\hat{y}) = \sum_j \alpha_{\hat{f}_j} > \frac{2}{3}k$. Furthermore, due to our assumption that $\epsilon_N > 1/4$, we know that $v'(N) > \frac{4}{3}v(N) > \frac{8}{9}k$. \square

As we have seen in the proof of Lemma 3.14, we may assume $S_2 \neq \emptyset$. The following result strengthens this observation:

Lemma 3.15. $|S_2| \geq \frac{2}{3}k - \frac{3}{4}$.

Proof. As in the proof of Lemma 3.14 we find

$$\begin{aligned} v'(N) &\leq v'(L) + \alpha_{S_1} + \alpha_{S_2}, \\ v(N) &\geq v(L) + \alpha_{S_1}. \end{aligned}$$

Thus, using Lemma 3.13, we get

$$v'(N) - v(N) \leq \alpha_{S_2} + \frac{1}{4} \leq \frac{|S_2|}{3} + \frac{1}{4}. \quad (3.20)$$

On the other hand, $\epsilon_N > 1/4$ and $v'(N) > \frac{8}{9}k$ (Lemma 3.14) imply

$$v'(N) - v(N) = \epsilon_N v'(N) > \frac{2}{9}k. \quad (3.21)$$

Combining (3.20) and (3.21), the bound on $|S_2|$ follows. \square

3.6.2 Greedy selection

Based on Lemma 3.15, we are left to deal with instances where Item Packing leaves a considerable amount $|S_2|$ of small items unpacked. As it turns out, a completely different kind of packing heuristic, so-called Greedy Selection, first analyzed in [45] is helpful in this case (also cf. Section 3.2). The basic idea is simple: We start with an optimum fractional packing y' of value $w(y') = v'(N)$. Let F'_1, \dots, F'_m denote the feasible sets in the support of y' (i.e., those that are fractionally packed) and assume that $\alpha_{F'_1} \geq \dots \geq \alpha_{F'_m}$. Greedy Selection starts with $F_1 := F'_1$ and then, after having selected F_1, F_2, \dots, F_{j-1} , defines F_j to be the first set in the sequence F'_1, \dots, F'_m that is disjoint from $F_1 \cup \dots \cup F_{j-1}$. It is not difficult to see (cf. below) that we can select disjoint feasible sets F_1, \dots, F_r with $r = \lceil k/3 \rceil$ in this way.

GREEDY SELECTION

Input: Optimal fractional packing y' with $\text{supp } y' = \{F'_1, \dots, F'_m\}$ and $\alpha_{F'_1} \geq \dots \geq \alpha_{F'_m}$.

Initiate: $\mathcal{F}'_1 := \{F'_1, \dots, F'_m\}$.

For $j = 1$ to $r = \lceil k/3 \rceil$ do

$F_j \leftarrow$ first set in the list \mathcal{F}'_j ;

$\mathcal{F}'_{j+1} \leftarrow \mathcal{F}'_j \setminus \{F'_t \mid F'_t \cap F_j \neq \emptyset\}$;

End

Note that Greedy Selection starts with $\mathcal{F}' = \text{supp } y'$, a system of feasible sets of total length $y'_{\mathcal{F}'} := \sum_{F' \in \mathcal{F}'} y'_{F'} = k$. In each step, since $|F_j| \leq 3$, we remove feasible sets F'_t of total length $\sum_{F'_t \in \mathcal{F}'_j, F'_t \cap F_j \neq \emptyset} y'_{F'_t}$ bounded by

$$\sum_{\substack{F'_t \in \mathcal{F}'_j \\ F'_t \cap F_j \neq \emptyset}} y'_{F'_t} \leq \sum_{i \in F_j} \sum_{\substack{F'_t \in \mathcal{F}' \\ F'_t \ni i}} y_{F'_t} \leq \sum_{i \in F_j} 1 \leq 3.$$

Actually, the upper bound 3 is strict since $F'_t = F_j$ is counted $|F_j|$ times (once for each $i \in F_j$).

Thus, in each step we remove feasible sets of total length at most 3, so we certainly can continue the construction for $\lceil k/3 \rceil$ steps.

Lemma 3.16. *Greedy Selection constructs feasible sets $F_1, \dots, F_r \in \mathcal{F}'$ with $r = \lceil k/3 \rceil = \frac{1}{3}(k+s)$, $k+s \equiv 0 \pmod{3}$, of total value $\alpha_{F_1} + \dots + \alpha_{F_r} \geq \frac{1}{3}(v'(N) + \frac{2}{3}s)$.*

Proof. The first part has been argued already above. To prove the lower bound, we first prove

Claim 1: $\alpha_{F_r} > \frac{2}{3}$.

Proof of Claim 1. Assume to the contrary that $\alpha_{F_r} \leq 2/3$. In the constructive process of Greedy Selection, when we have selected F_1, F_2, \dots, F_{r-1} , the remaining feasible set system \mathcal{F}'_r has still length larger or equal to $k - 3(r-1) = 3 - s \geq 1$. As F_r has maximum size (value) among all sets in \mathcal{F}'_r , we know that each set in \mathcal{F}'_r has size $\leq 2/3$. We now decrease y' on \mathcal{F}'_r such that the resulting fractional packing \tilde{y}' has total length $\sum_F \tilde{y}'_F = k - 1$. This decrease can be done in an arbitrary way. By construction, the corresponding game \tilde{N} with one bin removed has fractional value

$$v'(\tilde{N}) \geq w(\tilde{y}') \geq w(y') - \frac{2}{3} = v'(N) - \frac{2}{3}.$$

By minimality of our counterexample N , the game \tilde{N} admits an integral packing \tilde{y} of value

$$v(\tilde{N}) = w(\tilde{y}) \geq \frac{3}{4}v'(\tilde{N}) \geq \frac{3}{4}(v'(N) - \frac{2}{3}) = \frac{3}{4}v'(N) - \frac{1}{2}.$$

Extending this packing \tilde{y} by filling the k^{th} bin to at least $1/2$ in a simple greedy manner (order not yet packed items non-increasingly in size and try to pack them into bin k in this order) yields an integral packing y for N of value

$$w(y) \geq w(\tilde{y}) + \frac{1}{2} \geq \frac{3}{4}v'(N),$$

contrary to our assumption on N . This finishes the proof of Claim 1.

To prove the bound on $\alpha_{F_1} + \dots + \alpha_{F_r}$ in Lemma 3.16, let $\mathcal{R}'_j \subseteq \mathcal{F}'_j$ denote those feasible sets that are removed from \mathcal{F}'_j in step j , i.e., $\mathcal{R}'_j =$

$\mathcal{F}_j \setminus \mathcal{F}'_{j+1}$. As we have seen, \mathcal{R}'_j has total length $y'_{\mathcal{R}'_j} = \sum_{F' \in \mathcal{R}'_j} y'_{F'} \leq 3$. Assume that we, in addition, also decrease the y' -value on the least valuable sets in \mathcal{F}'_r by a total of $3 - y'_{\mathcal{R}'_j}$ in each step. Thus we actually decrease the length of y' by exactly 3 in each step without any further impact on the construction. The total amount of value removed this way in step j is bounded by $3a_{F_j}$. If $k \equiv 0 \pmod 3$, i.e. $s = 0$, we remove in all $r = k/3$ rounds exactly $v'(N)$. Thus

$$v'(N) \leq \sum_{j=1}^r 3a_{F_j},$$

as claimed.

When $k \not\equiv 0 \pmod 3$, the same procedure yields a reduced length of \mathcal{F}'_r after $r - 1$ steps, namely $y'_{\mathcal{F}'_r} = k - 3(r - 1) = 3 - s$. So in the last step we remove a set \mathcal{R}'_r of value at most $(3 - s)a_{F_r} \geq \frac{2}{3}(3 - s) = 2 - \frac{2}{3}s$ in the last step. Summarizing, the total value removed is

$$\begin{aligned} v'(N) &\leq 3a_{F_1} + \cdots + 3a_{F_{r-1}} + (3 - s)a_{F_r} \\ &= 3(a_{F_1} + \cdots + a_{F_r}) - sa_{F_r} \\ &\leq 3(a_{F_1} + \cdots + a_{F_r}) - \frac{2}{3}s, \end{aligned}$$

proving the claim. □

3.6.3 The combination – set packing

A natural idea is to extend the greedy selection F_1, \dots, F_r in a straightforward manner to an integral packing $y \hat{=} (F_1, \dots, F_k)$, where F_{r+1}, \dots, F_k are determined by applying Item Packing to the remaining items and remaining $k - r$ bins. However, this does not necessarily yield a sufficiently high packing value $w(y)$: Indeed, it may happen that the remaining $k - r \approx \frac{2}{3}k$ bins get only filled to roughly $1/2$, so the total packing value is approximately $w(y) \approx \frac{1}{3}v'(N) + \frac{1}{2} \cdot \frac{2}{3}k$, which equals $\frac{2}{3}v'(N)$ in case $v'(N) \approx k$.

However, the estimate $a_{F_{r+1}} + \cdots + a_{F_k} \approx \frac{1}{2}(k - r)$ is rather pessimistic. In particular, if we assume that the packing y (obtained by

Greedy Selection plus Item Packing the remaining $k - r$ bins) leaves some small items unpacked, then each of the $k - r$ bins must necessarily be filled to at least $2/3$ (otherwise any small item could be added on top of y). This would yield

$$w(y) \approx \frac{1}{3}v'(N) + \frac{2}{3}k \cdot \frac{2}{3} \geq \left(\frac{1}{3} + \frac{4}{9}\right)k \geq \frac{7}{9}k \geq \frac{7}{9}v'(N),$$

which is certainly sufficient for our purposes. (We do not use this estimate later on: It is only meant to guide our intuition.) Thus the crucial instances are those where y packs all small items – and hence does not pack all large items. Thus, when Item Packing is performed on the $k - r$ remaining bins, the first phase will pack large items into these $k - r$ bins until each bin is at least filled to $1/2$. Packing small items on top of any such bin would result in a bin filled to at least $3/4$. Thus, again, the worst case appears to happen when all small items (and there are quite a few of these, *cf.* Lemma 3.15) are already contained in $F_1 \cup \dots \cup F_r$. Assume for a moment that each of F_1, \dots, F_r ($r \approx k/3$) contains two of the small items in S_2 . (Recall that $|S_2| \gtrsim \frac{2}{3}k$.) Then each of F_1, \dots, F_r can contain only one other item in addition. Thus about $\frac{2}{3}k$ of the sets \hat{F}_j computed by Item Packing must be disjoint from $F_1 \cup \dots \cup F_r$ (as no \hat{F}_j contains any item from S_2). Thus we could extend F_1, \dots, F_r by roughly $\frac{2}{3}k \approx k - r$ sets, say, $\hat{F}_{r+1}, \dots, \hat{F}_k$ from Item Packing to obtain a packing $(F_1, \dots, F_r, \hat{F}_{r+1}, \dots, \hat{F}_k)$ of value $\geq \frac{1}{3}v'(N) + \frac{2}{3}k \cdot \frac{2}{3}$ (as each \hat{F}_j has size $\geq \frac{2}{3}$), which is again enough.

In general, the Greedy Selection F_1, \dots, F_r will contain some – but not all – items of S_2 and we will have to work out a trade-off between the two extreme cases considered above: Let

$$\gamma := |S_2 \cap (F_1 \cup \dots \cup F_r)|.$$

Thus $F_1 \cup \dots \cup F_r$ contains at most $3r - \gamma$ items that are not in S_2 – and hence possibly contained in some \hat{F}_j . So there are at most $3r - \gamma$ different \hat{F}_j that intersect $F_1 \cup \dots \cup F_r$ – and hence at least $k - (3r - \gamma)$ different \hat{F}_j that do not intersect $F_1 \cup \dots \cup F_r$. We add these $k - 3r + \gamma = \gamma - s$ different \hat{F}_j , say, $\hat{F}_{r+1}, \dots, \hat{F}_{r+\gamma-s}$ to F_1, \dots, F_r to obtain a

“partial” packing $F_1, \dots, F_r, \hat{F}_{r+1}, \dots, \hat{F}_{r+\gamma-s}$. Here we may assume that $r + \gamma - s < k$. Otherwise, we finish with a complete packing $\bar{y} = F_1, \dots, F_r, \hat{F}_{r+1}, \dots, \hat{F}_k$ of value (use Lemma 3.16 and $\alpha_{\hat{F}_j} \geq 2/3$).

$$\begin{aligned} w(\bar{y}) &\geq \frac{1}{3}(v'(N) + \frac{2}{3}s) + (k-r) \cdot \frac{2}{3} \\ &= \frac{1}{3}v'(N) + \frac{2}{9}s + (\frac{2}{3}k - \frac{1}{3}s) \frac{2}{3} \\ &= \frac{1}{3}v'(N) + \frac{4}{9}k \\ &\geq (\frac{1}{3} + \frac{4}{9})v'(N) \\ &> \frac{3}{4}v'(N), \end{aligned}$$

contrary to our assumption that $v(N) < \frac{3}{4}v'(N)$.

Thus $r + \gamma - s < k$ holds indeed and therefore we may complete our partial packing $F_1, \dots, F_r, \hat{F}_{r+1}, \dots, \hat{F}_{r+\gamma-s}$ by Item Packing the remaining items to the remaining $k - (r + \gamma - s)$ bins, resulting in an integral packing

$$\bar{y} \hat{=} F_1, \dots, F_r, \hat{F}_{r+1}, \dots, \hat{F}_{r+\gamma-s}, F_{r+\gamma-s+1}, \dots, F_k.$$

This completes the description of our heuristic

SET PACKING

1. Get F_1, \dots, F_r from Greedy Selection and let

$$\gamma := |S_2 \cap (F_1 \cup \dots \cup F_r)|.$$

2. Extend with sets \hat{F}_j from Item Packing to

$$F_1, \dots, F_r, \hat{F}_{r+1}, \dots, \hat{F}_{r+\gamma-s}.$$

3. Complete by Item Packing to

$$F_1, \dots, F_r, \hat{F}_{r+1}, \dots, \hat{F}_{r+\gamma-s}, F_{r+\gamma-s+1}, \dots, F_k.$$

Lemma 3.17. *Set Packing packs all items of S (and hence not all items of L).*

Proof. If Set Packing leaves some small item unpacked, then necessarily $\alpha_{F_j} > 2/3$ for $j = r + \gamma - s + 1, \dots, k$. Thus the same computation as above yields

$$w(\bar{y}) \geq \frac{1}{3}(v'(N) + \frac{2}{3}s) + (k-r) \frac{2}{3} > \frac{3}{4}v'(N),$$

a contradiction. Thus all of items S get packed. If, in addition, all items of L would get packed, then the value of the resulting packing \bar{y} is $w(\bar{y}) \geq a_S + a_L \geq v'(N)$, which again is a contradiction. \square

Thus, in phase 3 of Set Packing, when we apply Item Packing to the last $k - (r + \gamma - s)$ bins, each bin gets first filled to at least $1/2$ by large items, and then (possibly among other small items), the remaining $|S_2| - \gamma$ items from S_2 get packed on top of (some of) the last $k - (r + \gamma - s)$ bins. So these last $k - (r + \gamma - s)$ bins contribute at least

$$\frac{1}{2}(k - (r + \gamma - s)) + \frac{1}{4}(|S_2| - \gamma)$$

to the total value of \bar{y} .

Summarizing, $w(\bar{y})$ can be estimated as

$$w(\bar{y}) \geq \frac{1}{3}(v'(N) + \frac{2}{3}s) + (\gamma - s)\frac{2}{3} + \frac{1}{2}(k - (r + \gamma - s)) + \frac{1}{4}(|S_2| - \gamma). \quad (3.22)$$

In case $k \equiv 0 \pmod{3}$, we have $s = 0$, $r = k/3$ and $|S_2| \geq \frac{2}{3}k$ (by Lemma 3.15). So (3.22) simplifies to

$$w(\bar{y}) \geq \frac{1}{3}v'(N) + \frac{1}{2} \cdot \frac{2}{3}k + \frac{1}{4} \cdot \frac{2}{3}k - \frac{1}{12}\gamma \geq \frac{3}{4}v'(N) + \frac{1}{12}k - \frac{1}{12}\gamma,$$

proving that $w(\bar{y}) \geq \frac{3}{4}v'(N)$ if $\gamma \leq k$. But, as we have already argued above, we may even assume that $r + \gamma - s = r + \gamma < k$. So this is true. (Recall that we consider the case $s = 0$ here.)

Next assume $k \not\equiv 0 \pmod{3}$. In this case, a similar computation, using $|S_2| \geq \frac{2}{3}k - \frac{3}{4}$ from Lemma 3.15, yields

$$\begin{aligned} w(\hat{y}) &\geq \frac{3}{4}v'(N) + \frac{1}{12}k - \frac{1}{12}\gamma + \left(\frac{1}{3} \cdot \frac{2}{3} - \frac{2}{3} + \frac{1}{2}\right)s - \frac{1}{4} \cdot \frac{3}{4} \\ &= \frac{3}{4}v'(N) + \frac{1}{12}k - \frac{1}{12}\gamma + \frac{1}{18}s - \frac{3}{16} \\ &\geq \frac{3}{4}v'(N) + \frac{1}{12}(k - \gamma + s) - \left(\frac{1}{12} - \frac{1}{18}\right)s - \frac{3}{16}. \end{aligned}$$

Now recall that $r + \gamma - s < k$, thus $k - \gamma + s > r = \lceil k/3 \rceil$. The cases $k = 1$ and $k = 2$ are trivial (as remarked earlier at the beginning of

Section 3.6), hence $k \not\equiv 0 \pmod{3}$ implies $k > 3$. Thus actually $k - \gamma + s \geq 3$, so that our estimate above yields

$$\begin{aligned} w(\hat{y}) &\geq \frac{3}{4}v'(N) + \frac{1}{4} - \frac{3}{16} - \left(\frac{1}{12} - \frac{1}{18}\right)s \\ &\geq \frac{3}{4}v'(N) + \frac{1}{16} - \left(\frac{1}{12} - \frac{1}{18}\right)2 \\ &= \frac{3}{4}v'(N) + \frac{1}{16} - \frac{1}{18} > \frac{3}{4}v'(N). \end{aligned}$$

Summarizing, both cases ($k \equiv 0$ and $k \not\equiv 0 \pmod{3}$) yield that N cannot be a counterexample, so we have proved

Theorem 3.18. *The 1/4-core of the uniform bin packing game is always nonempty.*

We like to note that – even though our proof is indirect – it can easily be turned into a constructive proof. Indeed, we implicitly show that either Item Packing or Set Packing yields an integral packing y of value $w(y) \geq \frac{3}{4}v'(N)$. Also note that an optimum fractional packing y' (as input to Greedy Selection) is efficiently computable: Indeed, as any feasible set may contain at most 3 items, the total number of feasible sets is $O(n^3)$, so the LP for computing y' has polynomial size. Thus, in particular, our approach also yields a strengthening of the result in [11] (efficient 3/4 approximation).

3.7 REMARKS AND OPEN PROBLEMS

Analyzing LP-relaxations and the resulting integrality gap has been a standard issue in combinatorial optimization since long. In recent years, the theoretical analysis of integrality gaps combined with (randomized) rounding techniques has led to interesting results in on-line algorithms (c.f. [5]) as well as approximation theory. Other related work aims at approximating the optimum packing value directly without regarding the fractional value. In particular, Caprara *et al.* [11] showed how to compute an integral packing (say y) of value

$w(y) \geq \frac{3}{4}v(N)$ in polynomial time. The relation to our results is rather unclear, as $w(y) \geq \frac{3}{4}v(N)$ does not even imply, say, $w(y) \geq \frac{2}{3}v'(N)$ in general.

The main contribution of this chapter is an improvement of the best known result on the non-emptiness of the ϵ -core from the bound $1/3$ (due to Woeginger [74]) to the bound $1/4$. Two packing approaches that may achieve the $1/4$ bound were introduced, however, it turns out that each approach alone has its own disadvantages (despite its advantages), which stand in the way of our purpose. Finally, we succeed in combining ideas from the previous approaches, however, with slight variants. Applying a new analysis, which is more subtle and elegant, to the new approach, we prove the claim.

Clearly the most straightforward open problem is to determine the smallest ϵ such that the bin packing game has a non-empty ϵ -core for any instance. In [45] it was conjectured that $\epsilon_N \leq 1/7$ is true. (This bound would be tight as can be seen from Example 3.1.) We do not expect that our arguments provide any clue about how to approach $1/7$.

A probably even more challenging conjecture due to Woeginger states that the integrality gap $\text{gap}(N) = v'(N) - v(N)$ is bounded by a constant. Until now, this has only been verified for item sizes $> 1/3$ (cf. Lemma 3.13). It would be interesting to investigate the case of item sizes $a_i > 1/4$. The largest gap found (cf. Joosten [40]) so far is $\text{gap}(N) \approx 1/3$, for a game with 6 bins and 18 items (cf. Example 3.19).

We finally would like to draw the attention of the reader to the well-known 3-Partition problem (cf. [31]): Given a set of items of sizes a_1, \dots, a_{3k} with $1/4 < a_i < 1/2$ and k bins, can we pack all items? If the fractional optimum is less than k , the answer is clearly “no”. Note that the fractional optimum can be computed efficiently as there are only $O(k^3)$ feasible sets. Thus if $P \neq NP$, then there must be instances with fractional optimum equal to k and integral optimum strictly less than k . Eventually, Joosten [40] succeeded in computing (probably the smallest) such instances. One of these is described in example 3.19 below.

Example 3.19 ([40]). Given six bins of capacity $b = 92$ each and 18 items of sizes: 0, 6, 10, 11, 12, 13, 14, 17, 18, 26, 29, 36, 50, 56, 57, 64, 66, 67.

Note that the above instance can be easily transformed to a standard 3-Partition instance by letting $a'_i = a_i + M$, $b' = b + 3M$, for sufficiently large M , say $M = 1000$. Normalizing the bin capacity and the item sizes, we get $a'_i \approx 1/3$ for $i = 1, \dots, 18$.

The optimal fractional packing is described in the following picture: Each node represents a feasible set, containing items represented by its adjacent edges, and each feasible set has a fraction $1/2$.

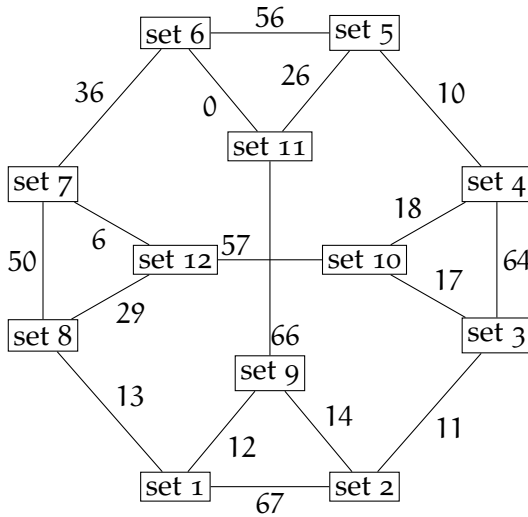


Figure 3.10: An instance with gap $\approx 1/3$

In the integral optimum packing, at least 1 item cannot be packed, resulting in a gap $\approx 1/3$.

Non-uniform bin packing game

It is natural to extend the uniform bin packing game that is studied in Chapter 3 to the non-uniform case, where bins may have different capacities. Faigle and Kern [27] first studied this problem and observed that the $1/2$ -core is always nonempty, provided that each item fits into each bin. However, results for the general (non-uniform) bin packing game are quite poor.

In this chapter, we investigate the relative integrality gap for the non-uniform bin packing game. Apparently, the problem becomes more difficult when capacities of bins are distinct. In particular, for the special case of large items ($a_i > 1/3$ for all i), the matching approach used in Section 3.4 cannot be applied any more, as in the non-uniform case, each bin may correspond to a different collection of feasible sets. New ideas are needed even in analyzing this particular case (*cf.* Section 4.3).

Based on our study in Chapter 3, some questions naturally arise, *e.g.*, whether the $1/2$ -core is nonempty, whether the instance with large item sizes ($a_i > 1/3$ for all i , where all bin capacities are smaller or equal to 1) is also easy to analyze, etc. In Section 4.1 below, we first derive an integer linear program for the non-uniform bin packing game. In Section 4.2, we show that the $1/2$ -core is always non-empty (without any assumption on item sizes). In Section 4.3, it turns out that even if we consider large item sizes, analyzing the relative inte-

grality gap is still difficult. We show that the $5/12$ -core is nonempty in this case. In Section 4.4, the limiting case $k \rightarrow \infty$ is considered. We extend the result in the uniform case (cf. [28]), stating that given a fixed $\epsilon > 0$ the ϵ -core is always nonempty if k is sufficiently large, to the non-uniform case. However, in the non-uniform case the average bin size must be taken into account. In the final Section 4.5, we remark on some open problems that might be interesting for future research.

4.1 ILP FORMULATION

Let us denote by b_1, b_2, \dots, b_k the capacities of bins $1, 2, \dots, k$ respectively and assume that the bins are ordered decreasingly, *i.e.*,

$$1 = b_1 \geq b_2 \geq \dots \geq b_k.$$

Most of the terminology and definitions from the uniform bin packing game carry over to the non-uniform setting except for the concept of feasible sets, which must now be related to individual bins. A set $F \subseteq A$ is called *feasible* for bin j , if the total size of items of F does not exceed the bin capacity b_j . Denote by \mathcal{F} the collection of all feasible sets and \mathcal{F}_j the collection of feasible sets for bin j , $j = 1, \dots, k$, thus,

$$\mathcal{F} = \mathcal{F}_1 \supseteq \mathcal{F}_2 \supseteq \dots \supseteq \mathcal{F}_k.$$

As before, given a set of items, say F , denote by a_F the total size of F , *i.e.*, $a_F = \sum_{i \in F} a_i$. Let $\mathcal{F}_{k+1} = \emptyset$ and define variables $y_F \in \{0, 1\}$ for all $F \in \mathcal{F}$ indicating whether F is packed. Hence, it is not hard to derive the following integer linear program for solving the optimum packing value $v(N)$,

$$\begin{aligned} & \text{maximize} && \sum_{F \in \mathcal{F}} a_F y_F, && (4.1) \\ & \text{subject to} && \sum_{F \ni i, F \in \mathcal{F}} y_F \leq 1, && (i = 1, \dots, n), \\ & && \sum_{F \in \mathcal{F} \setminus \mathcal{F}_{j+1}} y_F \leq j, && (j = 1, \dots, k), \end{aligned}$$

$$y_F \in \{0, 1\}, \quad \text{for all } F \in \mathcal{F}.$$

The objective and the first set of constraints are easy to understand. The second set of constraints guarantees that each bin packs feasible sets from its corresponding feasible sets. Note the inequality makes use of the (non-increasing) ordering of bin sizes.

As before, we consider the relaxation of (4.1) and introduce the fractional packing.

$$\begin{aligned} & \text{maximize} && \sum_{F \in \mathcal{F}} a_F y_F, && (4.2) \\ & \text{subject to} && \sum_{F \ni i, F \in \mathcal{F}} y_F \leq 1, && (i = 1, \dots, n), \\ & && \sum_{F \in \mathcal{F} \setminus \mathcal{F}_{j+1}} y_F \leq j, && (j = 1, \dots, k), \\ & && y_F \geq 0, && \text{for all } F \in \mathcal{F}. \end{aligned}$$

A feasible solution to (4.2) is called a *fractional packing*. It is not difficult to see that the above linear program corresponds to (2.2) (cf. Section 2.2). Let $v'(N)$ be the optimal objective function value of (4.2). By Lemma 2.2, the ϵ -core is nonempty if and only if $\epsilon \geq 1 - v(N)/v'(N)$. Therefore, the minimal taxation rate is $\epsilon_N := 1 - v(N)/v'(N)$, as before. In the following sections, we will analyze the relative integrality gap $v(N)/v'(N)$ for the non-uniform bin packing game.

4.2 NON-EMPTINESS OF THE 1/2-CORE

To see the non-emptiness of the 1/2-core for the non-uniform bin packing game is not that simple as in the uniform case. Recall the proof (of 1/2-core $\neq \emptyset$) in the uniform case, an integral packing with a total value at least $k/2$ can be obtained via a straightforward greedy packing, and the claim follows immediately. However, in the non-uniform case, a bin is not guaranteed to be filled to its half capacity, because items may be too large to be packed into that bin.

We first present a simple packing algorithm for constructing an integral packing: Consider a bin b_j (with slight abuse of notation, we also refer to b_j as bin j if no confusion occurs) and a set $\{a'_1, \dots, a'_s\}$ of items that fit into b_j (i.e., $a'_i \leq b_j$). The simple packing algorithm either packs all items into b_j (if $\sum_i a'_i \leq b_j$) or computes a subset A' of items that has total size $a_{A'} \geq \frac{1}{2}b_j$:

SIMPLE PACKING

Input: bin b_j , items $a'_1, \dots, a'_s \leq b_j$.
 If $\sum_i a'_i \leq b_j$ Then return $\{a'_1, \dots, a'_s\}$;
 Else
 Let $a'_1 + \dots + a'_r \leq b_j$, $a'_1 + \dots + a'_{r+1} > b_j$;
 Return the larger of $\{a'_1, \dots, a'_r\}$ and $\{a'_{r+1}\}$;
 End

The simple packing algorithm can be used to derive a packing heuristic, constructing an integer packing for N : Let $A_j \subseteq A$ denote the set of items that fit into b_j . We first apply simple packing to b_k and A_k . Assume that the simple packing algorithm packs $F_k \subseteq A_k$ into b_k . We then apply simple packing to b_{k-1} and $A_{k-1} \setminus F_k$ and assume that $F_{k-1} \subseteq A_{k-1} \setminus F_k$ gets packed into b_{k-1} etc. Continuing this way, we find

Lemma 4.1. *The simple packing heuristic computes an integral packing F_1, \dots, F_k such that either*

- (i) $a_{F_j} \geq \frac{1}{2}b_j$ for $j = 1, \dots, k$ or
- (ii) $a_{F_j} \geq \frac{1}{2}b_j$ for $j = 1, \dots, r$ and $F_{r+1} \cup \dots \cup F_k = A_{r+1}$ for a suitable $0 \leq r < k$ (possibly $r = 0$).

Proof. Assume that by applying the simple packing heuristic as described above, never a situation occurs that all “remaining” items

$$A_j \setminus (F_{j+1} \cup \dots \cup F_k)$$

fit into b_j . Then each bin gets filled to at least half its capacity (by simple packing). Otherwise, (ii) follows by letting r denote the smallest

j such that indeed all “remaining” items were packed into b_{j+1} , and hence all of A_{j+1} was packed into bins b_{j+1}, \dots, b_k . \square

As a simple consequence, we obtain the following.

Theorem 4.2. $\frac{1}{2}$ -core(N) $\neq \emptyset$ for all N.

Proof. Let $v(N), v'(N)$ denote the optimal integral and fractional packing value respectively. Clearly, $v(N) \geq a_{F_1} + \dots + a_{F_k}$ for the simple packing F_1, \dots, F_k . Thus, in case (i) of Lemma 4.1, we readily find

$$v(N) \geq a_{F_1} + \dots + a_{F_k} \geq \frac{1}{2}(b_1 + \dots + b_k) \geq \frac{1}{2}v'(N),$$

and the claim follows. If case (ii) occurs, then all of A_{r+1} gets packed into b_{r+1}, \dots, b_k by the simple packing heuristic. As a consequence, we find that the game N naturally splits into

$$N^{\text{red}} := (\{b_1, \dots, b_r\}, A \setminus A_{r+1})$$

and

$$N^{\text{triv}} := (\{b_{r+1}, \dots, b_k\}, A_{r+1}).$$

Indeed, as no item in $A \setminus A_{r+1}$ fits into any bin b_j , $j \geq r+1$, an optimum fractional packing y' for N can assign items in $A \setminus A_{r+1}$ only to bins b_1, \dots, b_r . As all of A_{r+1} can be packed (even integrally) into b_{r+1}, \dots, b_k , an optimum fractional packing y' can be assumed to fractionally pack part of $A \setminus A_{r+1}$ into b_1, \dots, b_r and all of A_{r+1} into b_{r+1}, \dots, b_k . Thus,

$$v'(N) = v'(N^{\text{red}}) + a_{A_{r+1}},$$

where v'_{red} is the fractional packing value for N^{red} . By Lemma 4.1, the simple packing heuristic yields a value

$$v(N) \geq \frac{1}{2}v'_{\text{red}} + a_{A_{r+1}} \geq \frac{1}{2}v'(N)$$

and the result follows. \square

We refer to N^{red} as defined in the proof of Theorem 4.2 as a *reduced game*. More generally, let us call N *reducible* if, for suitable $r \leq k$, all items in A_{r+1} can be (integrally) packed into b_{r+1}, \dots, b_k . Thus, as we have seen in the proof of Theorem 4.2, reducible games inherit ϵ -balancedness from their corresponding reductions $N^{\text{red}} = (\{b_1, \dots, b_r\}, A \setminus A_{r+1})$.

4.3 LARGE ITEMS: $a_i > 1/3$

In the uniform case, instances with large items $a_i > 1/3$ have attracted much attention. In practice, such instances may occur in large express firms which only deal with large goods, *i.e.*, small items are not delivered by them (as delivering small items yields less profits and causes almost the same administration costs). In theoretical terms, the case $a_i > 1/3$ is critical for proving non-emptiness of the $1/3$ -core. A standard proof technique for showing non-emptiness of the $1/3$ -core in the uniform case works as follows (*cf.* Section 3.1.3): First reduce the problem to the case where all items have size strictly larger than $1/3$. In these reduced problem instances, at most two items fit into a bin. Hence a fractional packing is close to a fractional matching of items and can thus be treated with well-known techniques from matching theory (*cf.* Section 3.4). In the non-uniform case, this approach does not work, as we shall explain below. Indeed, it is even unclear whether (4.2) always has an optimal solution that is half-integral. (In the uniform case, this follows quite easily by standard arguments from fractional matching theory, *cf.*, *e.g.*, [28].)

Even though the reduction (to MWM) does not work, yet we want to know whether small items affect the ϵ -balancedness. We answer this question in the section below, which might be of independent interest.

4.3.1 On small items

Still, the reduction to large item sizes can be extended to the non-uniform case. As it turns out, in the non-uniform case we have to distinguish between small and large items, where “small” and “large” are defined relative to the *average bin size* $\bar{b} := \sum_{j=1}^k b_j/k$. This can be seen from the following lemma.

Lemma 4.3. *Let N be a bin packing game and assume N is ϵ -balanced for some $\epsilon < 1/2$. Then adding “small” items of size $\alpha_i \leq \epsilon \bar{b}$ does not affect ϵ -balancedness.*

Proof. First note that it suffices to prove the claim in case where a single small item i_0 is added. Let $N^+ := N \cup \{\alpha_{i_0}\}$ denote the extended game. We consider an optimum integral packing y^* for N and distinguish two cases:

Case 1: The new item i_0 can be packed on top of the optimum integral packing for N (i.e. some bin j is filled only up to at most $b_j - \alpha_{i_0}$). In this case, we conclude that $v(N^+) = v(N) + \alpha_{i_0}$, whereas clearly, $v'(N^+) \leq v'(N) + \alpha_{i_0}$ (Take an optimal fractional packing for N^+ and remove item α_{i_0} from each feasible set to obtain a corresponding feasible fractional packing for N). Hence, $\epsilon_{N^+} \leq \epsilon_N \leq \epsilon$ follows.

Case 2: In the optimum integral packing for N , each bin with capacity $b_j \geq \epsilon \bar{b}$ is filled to more than $b_j - \alpha_{i_0}$. In this case, the total content of each bin is at least $b_j - \epsilon \bar{b}$, hence

$$\frac{v(N^+)}{v'(N^+)} \geq \frac{v(N)}{\sum_{j=1}^k b_j} \geq \frac{\sum_{j=1}^k (b_j - \epsilon \bar{b})}{\sum_{j=1}^k b_j} = \frac{k\bar{b} - k\epsilon \bar{b}}{k\bar{b}} = 1 - \epsilon,$$

proving ϵ -balancedness of N^+ . □

Unfortunately, Lemma 4.3 is of not much help in simplifying matters: Indeed, by adding a number of small dummy bins (plus corresponding items if we like), the average bin size can be made arbitrarily small – and hence the item sizes become relatively large (compared to the average bin size \bar{b}) – without significant change in the instance.

If we instead restrict ourselves to item sizes that are large in an absolute sense, the bound $\epsilon \leq 1/2$ can be somewhat improved (although, as compared to the uniform case, with considerably more effort and weaker result, cf. Theorem 4.7).

4.3.2 Properties of the counterexample

Given a feasible integral/fractional packing y , as before, we let $w(y) := \sum_{F \in \mathcal{F}} a_F y_F$ denote the total value of y . Let y' be an optimum fractional solution with value $v'(N) = w(y')$. We seek to “round” y' to an integral packing y of value $w(y) \geq \frac{7}{12}v'(N)$. The method we use is a modification of the rounding technique introduced in Section 3.2.

To achieve this, similar to the proof of non-emptiness of the $1/4$ -core in Section 3.6, we assume that N is a minimal counterexample, i.e. the $5/12$ -core is empty and the total number of players is minimized. Then we seek to achieve some simplifications by induction on the total number of items. We always assume that an integral packing \tilde{y} of some instance \tilde{N} ($|\tilde{N}| < |N|$) having a value $w(\tilde{y}) \geq \frac{7}{12}v'(\tilde{N})$ in each induction, where the factor $7/12$ suffices to achieve our goal. It is also possible to attain these simplifications by assuming other factors (larger than $7/12$), however, this will not help us to improve the lower bound on the relative integrality gap.

Let $\mathcal{F}' = \{F_1, \dots, F_m\}$ denote the support of y' and assume that

$$a_{F_1} \geq \dots \geq a_{F_m}.$$

We think of F_1, \dots, F_m as being assigned to bins $b_1 \geq \dots \geq b_k$ in this order, so that every bin except possibly the last ones are assigned feasible sets of total y -value equal to 1. Thus a feasible set F_s may get assigned to two consecutive bins j and $j+1$ if $y_{F_1} + \dots + y_{F_{s-1}} < j$ and $y_{F_1} + \dots + y_{F_s} > j$.

Proposition 4.4. $y'_F < 1$ for all $F \in \mathcal{F}'$.

Proof. We proceed by induction on the number of players. If $y'_{F_j} = 1$ for some j , we remove all items contained in F_j and the bin to which

F_j is assigned. (If F_j is assigned to two bins, choose the smaller one.) Let \tilde{N} denote the resulting instance. Obviously, y' induces a feasible fractional packing \tilde{y}' for \tilde{N} of value $w(\tilde{y}') = v'(N) - \alpha_{F_j}$. By induction, there exists a corresponding integral packing \tilde{y} of value $w(\tilde{y}) \geq \frac{7}{12}v'(\tilde{N})$. Extend this integral packing to an integral packing for N by packing F_j into the removed bin. The resulting integral packing has value $w(\tilde{y}) + \alpha_{F_j} \geq \frac{7}{12}v'(\tilde{N}) + \alpha_{F_j} \geq \frac{7}{12}v'(N)$. \square

Thus, in what follows, we may (and will) assume that $y'_F < 1$ for all $F \in \mathcal{F}'$.

Proposition 4.5. *All item sizes are less than $2/3$.*

Proof. Assume to the contrary that some item has size $\alpha = \alpha_{\max} \geq 2/3$. Then α cannot be combined with any other item into a feasible set. Hence there must be a single-item set $F_s = \{\alpha\}$. (We tacitly assume that item α is used at all – otherwise the Theorem follows by induction on the number of items.) According to Proposition 4.4, we may assume $y'_{F_j} < 1$. We now remove the (smallest) bin, say, b_j , to which $F_s = \{\alpha\}$ is assigned, together with other feasible sets assigned to b_j so that the removed feasible sets have a y' -value of exactly 1. The resulting fractional packing \tilde{y} for the instance $\tilde{N} = N \setminus \{\alpha, b_j\}$ has value $w(\tilde{y}) \geq v'(N) - 1$ and, by induction, there exists an integral packing \hat{y} of value $w(\hat{y}) \geq \frac{7}{12}(v'(N) - 1)$. Adding item α filled into bin b_j , we obtain an integral packing for N of value $\geq \frac{7}{12}(v'(N) - 1) + \alpha$. Thus,

$$v(N) \geq \frac{7}{12}(v'(N) - 1) + \alpha \geq \frac{7}{12}(v'(N) - 1) + \frac{2}{3} > \frac{7}{12}v'(N).$$

\square

Thus, in what follows, we assume that Proposition 4.5 holds w.l.o.g.

Proposition 4.6. *At least one two-element set F_j is assigned to b_k . (Hence, in particular, $b_k > 2/3$.)*

Proof. According to Proposition 4.5, the one-element sets have smaller size than the two-element sets, and, hence, appear last in the ordering

F_1, \dots, F_m . Now assume that all sets F_m, F_{m-1} etc. assigned to b_k are one-element sets and let $F_m = \{a\}$. (If no F_j is assigned to b_k , the claim of the Theorem follows by induction on the number of bins.)

We first show that we may assume $\sum_{F \in \mathcal{F}'} y'_F = k$ w.l.o.g. Indeed, if $\sum_{F \in \mathcal{F}'} y'_F < k$, we first try to increase this sum by increasing y'_{F_m} as much as possible until either $\sum_{F \in \mathcal{F}'} y'_F = k$ holds (and we are done) or item a gets fully packed in the sense that $\sum_{F \ni a} y'_F = 1$. We then seek to increase y'_{F_m} further by *splitting* a suitable feasible set $F_j = \{a, a_i\}$, i.e., we increase both $y'_{\{a_i\}}$ and $y'_{\{a\}}$ and decrease y'_{F_j} by the same amount. Note that this modification keeps y' optimal, as basically F_j is replaced by the smaller feasible set $\{a_i\}$. Proceeding this way we eventually end up with a modified feasible fractional packing (which we again denote by y') of equal value $v'(N)$ that satisfies $\sum_{F \in \mathcal{F}'} y'_F = k$ (unless, in between, either $y'_{\{a\}}$ or $y'_{\{a_i\}}$ is increased to 1 and induction applies anyway). Thus, we may indeed assume $\sum_{F \in \mathcal{F}'} y'_F = k$ in the following.

As $\sum_{F \in \mathcal{F}'} y'_F = k$ holds, the total y' -value of sets assigned to b_k equals 1. Thus there are at least two one-element sets $F_m = \{a\}$ and $F_{m-1} = \{a'\}$, say, assigned to b_k (as we assume $y'_{F_m} < 1$). Since $\alpha_{F_{m-1}} \geq \alpha_{F_m}$, we have $a' \geq a$. We seek to reduce y'_{F_m} to 0. To this end, we first increase $y'_{F_{m-1}}$ and decrease y'_{F_m} as much as possible until either $y'_{F_m} = 0$ (and the claim follows by induction on the number of single-element sets in the support of y' – under the additional assumption that $\sum_{F \in \mathcal{F}'} y'_F = k$) or a' gets fully packed, i.e., $\sum_{F \ni a'} y'_F = 1$. In the latter case we seek to reduce y'_{F_m} further by replacing a' with a as much as possible in any set $F_j = \{a', a_i\}$ with $a_i \neq a$. More precisely, as long as there is some $F_j = \{a', a_i\}$ with $y'_{F_j} > 0$ and $a_i \neq a$, we decrease y'_{F_j} and y'_{F_m} and increase $y'_{\{a, a_i\}}$ and $y'_{\{a'\}}$ by the same amount. Note that this modification keeps y' feasible, since $a' \geq a$, so F_j is (partially) replaced by a smaller feasible set in the fractional packing. This modification stops when the only feasible two-element set containing a' is $F_j = \{a', a\}$. Note that, at that point of our modification, we have $y'_{\{a, a'\}} + y'_{\{a'\}} = 1$.

Assume for a moment that there is a third single element set $F_{m-2} = \{a''\}$ assigned to b_k with $a'' \geq a'$. We could then repeat the above modification w.r.t. a'' and a' (instead of a' and a), thereby either succeeding in reducing $y'_{F_{m-1}}$ to 0 (in which case induction on the number of single-element sets in the support of y' applies) or getting stuck in a situation where a'' is fully packed but the only two-element set containing a'' is $\{a'', a'\}$. But this would contradict our assumption that a' is only combined with a in a feasible set $F_j = \{a', a\}$.

Summarizing, we may assume that F_{m-2} is assigned to b_{k-1} and, consequently, $y'_{\{a'\}} + y'_{\{a\}} \geq 1$. Hence $y'_{\{a',a\}} + y'_{\{a'\}} = 1$ and $y'_{\{a',a\}} + y'_{\{a\}} \leq 1$ imply $y'_{\{a'\}} \geq y'_{\{a',a\}}$ and, therefore, $y'_{\{a'\}} \geq \frac{1}{2}$ and $y'_{\{a',a\}} \leq \frac{1}{2}$. Removing b_k with all its content and item a' from F_j (the only two-element set containing a') results in a feasible fractional packing \tilde{y}' for $\tilde{N} := N \setminus \{a', b_k\}$ of value

$$w(\tilde{y}') = v'(N) - a' - y'_{F_m} a \geq v'(N) - a' - \frac{1}{2}a \geq v'(N) - \frac{3}{2}a'$$

By induction, there is a corresponding integral solution \hat{y} of value $w(\hat{y}) \geq \frac{7}{12}v'(\tilde{N})$. Adding item a' (assigned to b_k), we obtain a packing for N of value $\geq \frac{7}{12}v'(\tilde{N}) + a'$. Thus,

$$v(N) \geq \frac{7}{12}v'(\tilde{N}) + a' \geq \frac{7}{12}(v'(N) - \frac{3}{2}a') + a' \geq \frac{7}{12}v'(N).$$

This completes the proof. \square

4.3.3 Reversed greedy packing

Having achieved the above three simplifications, we are now ready to proceed to the main part of the analysis, which consists in "rounding" y' to an integer packing y , with value $w(y) \geq \frac{7}{12}v'(N)$. The basic idea is a greedy selection rule similar to the one introduced in Section 3.2. The main difference is that, here, we construct pairwise disjoint feasible sets F_{j_1}, \dots, F_{j_r} in a reverse order, *i.e.*, starting with the *smallest* feasible two-element set F_j rather than with the largest (as we did in Section 3.2).

Theorem 4.7. *If all items have size $a_i > 1/3$, the 5/12-core is nonempty.*

Proof. We let $F_{j_1} \in \mathcal{F}$ denote the smallest two-element set in the support of y' (note that F_{j_1} is assigned to b_k), and choose it as a feasible set of our integral packing (*i.e.*, $y_{F_{j_1}} = 1$). Then we look for the next (two-element) set among F_{j_1-1}, \dots, F_1 that is disjoint from F_{j_1} and call it F_{j_2} *etc.*. Thus in each step we determine the smallest feasible set that is disjoint from all previously selected ones. As each of the selected feasible sets F_{j_ρ} contains exactly two items, say, $F_{j_\rho} = \{a_i, a_l\}$, the total y' -value of feasible sets intersecting F_{j_ρ} is bounded by $2 - y_{F_{j_\rho}}$, for $\rho = 1, \dots, r$. This is straightforward from

$$\sum_{F \cap F_{j_\rho} \neq \emptyset} y'_F \leq \sum_{F \ni a_i} y'_F + \sum_{F \ni a_l} y'_F - y'_{F_{j_\rho}} \leq 2 - y'_{F_{j_\rho}}. \quad (4.3)$$

For that reason, F_{j_1}, \dots, F_{j_r} can be assigned to bins $b_k, b_{k-2}, \dots, b_{k-2(r-1)}$ (in that order). Due to Proposition 4.6, the total fraction (*i.e.*, the total y' -value) of packed 2-element sets is larger than $k - 1$, summing (4.3) yields

$$\sum_{\rho=1}^r (2 - y'_{F_{j_\rho}}) > k - 1,$$

implying $2r \geq k$.

For the remaining $k - r$ bins, w.l.o.g., we assume that half capacity of each bin can be filled by greedily packing items to those bins (as $a_i < 2/3 < b_k$ for all i , *cf.* also [27] or apply the simple packing heuristic). Let R be the index set of the remaining $k - r$ bins and $\bar{b}(R)$ be the corresponding average bin size. In case k is even, we have $\bar{b}(R) \geq \bar{b}$. Hence,

$$\begin{aligned} v(N) &\geq \frac{2}{3}r + \sum_{j \in R} \frac{b_j}{2} = \frac{2}{3}r + (k - r) \frac{\bar{b}(R)}{2} \\ &\geq \frac{2}{3} \cdot \frac{k}{2} + \frac{k}{2} \cdot \frac{\bar{b}}{2} \geq \frac{v'(N)}{3} + \frac{v'(N)}{4} = \frac{7}{12}v'(N). \end{aligned}$$

For k odd, the approximation is even better as we have in addition b_1 filled to at least $\frac{2}{3}$ of its capacity. This completes the proof. \square

4.4 LIMITING CASE: $k \rightarrow \infty$

In this section, we seek to extend the result of Faigle and Kern [28], saying that the ϵ -core is non-empty provided the game is “large enough”, to the non-uniform case. As in [28], our arguments are based on the bin packing approach initially introduced by de la Vega and Lueker [23]. The idea works as follows: First consider the case when item sizes and bin sizes take on at most m distinct values respectively (for some fixed number m) and obtain an upper bound on the integrality gap (which is independent of k). Then we consider the general case by rounding down item sizes and bin sizes to m distinct values respectively (thereby applying the result derived in the first step) and show that the decrease of the fractional optimum is bounded by a small rounding error. By taking m large enough, the rounding error becomes fairly small and the assumption on the distinct values of items and bins thus can be removed. Finally, by Theorem 4.2 we can assume that each bin is half-filled. Taking $k \rightarrow \infty$, we prove the claim.

4.4.1 *Restricting item sizes and bin sizes*

Consider the class of the non-uniform bin packing game where the number of distinct item sizes and the number of distinct bin sizes are bounded by m . Assume that the item sizes are a_1, \dots, a_m and occur with multiplicities $\alpha_1, \dots, \alpha_m$, and assume that the bin sizes are b_1, \dots, b_m and occur with multiplicities β_1, \dots, β_m . Clearly, $k = \sum_{i=1}^m \beta_i$. The idea is to find an upper bound of $v'(N) - v(N)$ which is independent of m . Since we can assume $v(N) \geq \frac{1}{2}k\bar{b}$ (by Simple Packing to each bin), letting $k \rightarrow \infty$ leads to non-emptiness of the ϵ -core for any fixed $\epsilon \in (0, 1)$.

Now each feasible set $F \in \mathcal{F}$ can be described by its *type vector* $T = (t_1, \dots, t_m)$ indicating the number t_i of items of size a_i that occur in F . Let $a_T = \sum_{i=1}^m t_i a_i$ and let \mathcal{T} be the set of type vectors.

Moreover, for each bin size b_j , denote by \mathcal{T}_j the set of type vectors, with $a_T \leq b_j$ for all $T \in \mathcal{T}_j$. Hence,

$$\mathcal{T} = \mathcal{T}_1 \supseteq \mathcal{T}_2 \supseteq \cdots \supseteq \mathcal{T}_m.$$

Let $\mathcal{T}_{m+1} = \emptyset$. Now $v(N)$ and $v'(N)$ (corresponding to the optimal objective function value of (4.1) and (4.2) respectively) can be computed by the following (integer) linear programs.

$$\begin{aligned} & \text{maximize} && \sum_{T \in \mathcal{T}} a_T z_T, && (4.4) \\ & \text{subject to} && \sum_{T \in \mathcal{T} \setminus \mathcal{T}_{i+1}} z_T \leq \sum_{j=1}^i \beta_j, && (i = 1, \dots, m), \\ & && \sum_{T \in \mathcal{T}} t_i z_T \leq \alpha_i && (i = 1, \dots, m), \\ & && z_T \in \mathbb{N}, && \text{for all } T \in \mathcal{T}, \end{aligned}$$

and

$$\begin{aligned} & \text{maximize} && \sum_{T \in \mathcal{T}} a_T z_T, && (4.5) \\ & \text{subject to} && \sum_{T \in \mathcal{T} \setminus \mathcal{T}_{i+1}} z_T \leq \sum_{j=1}^i \beta_j, && (i = 1, \dots, m), \\ & && \sum_{T \in \mathcal{T}} t_i z_T \leq \alpha_i && (i = 1, \dots, m), \\ & && z_T \in \mathbb{R}^+, && \text{for all } T \in \mathcal{T}, \end{aligned}$$

Denote by $\text{gap}(N)$ the integrality gap of N (non-uniform bin packing game instance), *i.e.* $\text{gap}(N) = v'(N) - v(N)$. Let a_A, a_B be the total size of the item set A and the total capacity of the bin set B , respectively.

Lemma 4.8. *If the item sizes and bin sizes assume at most m different values, then $\text{gap}(N) \leq m$.*

Proof. Let $z^* = (z_T^*)_{T \in \mathcal{T}}$ be an optimal fractional packing which is a basic feasible solution of (4.5). As there are only $2m$ constraints in (4.5),

we conclude that $|\text{supp}(z^*)| \leq 2m$, where $\text{supp}(z^*) = \{z_T > 0, T \in \mathcal{T}\}$. Furthermore, we may assume that $z^* \leq 1$ (componentwise). Indeed, assume $z_T^* > 1$ and let b_j denote the smallest bin size to which a set of type T is assigned by z^* . Reducing the multiplicities of all items in T by 1 and, similarly, replacing β_j by $\beta_j - 1$, we obtain a modified instance \tilde{N} with fractional packing value $v'(N) - \alpha_T$ and, by induction, a corresponding integral packing of value $\geq v'(\tilde{N}) - m$. Extending this to an integral packing for N in the obvious way (by assigning a set of type T to a bin of type j), the claim follows. Thus we may indeed assume that $z^* \leq 1$, and hence

$$v'(N) = \sum_{T \in \mathcal{T}} \alpha_T z_T^* \leq \sum_{T \in \mathcal{T}} z_T^* \leq |\text{supp}(z^*)| \leq 2m.$$

Theorem 4.2 then implies

$$\frac{v'(N) - v(N)}{v'(N)} \leq \frac{1}{2},$$

i.e., $v'(N) - v(N) \leq v'(N)/2 \leq m$ and the claim follows. \square

Note that the upper bound m is independent of k (the total number of bins). In the following, we want to reduce N to this restricted case by rounding down item sizes and bin sizes respectively.

4.4.2 Rounding items and bins

The idea is to divide n items (ordered non-decreasingly) and k bins (ordered non-increasingly) into $m + 1$ consecutive sublists of items and bins respectively. Each of the first m sublists of items (bins) has $h = \lfloor n/m \rfloor$ elements and the last sublist of items (bins) contains the remaining items (bins). For any feasible fractional packing of N , we present a rounded (feasible fractional) packing for the modified instance, with rounding error at most $4h$ (note that h becomes very small when m is sufficiently large). Combining Lemma 4.8, we obtain the following result.

Lemma 4.9. *Let $\epsilon > 0$ be such that $\epsilon^{-1} \in \mathbb{N}$. Then $\alpha_B \geq \epsilon n$ (where n is the total number of items) implies $\text{gap}(\mathbb{N}) \leq \epsilon^{-2} + 4\epsilon\alpha_B$.*

Proof. Assume items are given by the following non-decreasingly ordered list,

$$A : \alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_n.$$

Given $m > 0$, $m \in \mathbb{N}$ and $h = \lfloor n/m \rfloor$, divide A into $m + 1$ consecutive sublists

$$A = A_1, \dots, A_m, R$$

satisfying $|A_i| = h$, $i = 1, \dots, m$ and $|R| < h$. Let α_{i_j} be the first element of A_j . We consider the modified item list

$$A^- = A_1^-, \dots, A_{m-1}^-, A_m^-, R,$$

where the sublist $A_j^- = \alpha_{i_j}, \dots, \alpha_{i_j}$ arises from A_j by replacing each element of A_j with a copy of the smallest item in the sublist.

On the one hand, any feasible (integral) packing relative to A^- yields a feasible (integral) packing of A if we replace elements of A_j^- by the corresponding elements of A_{j-1} , for $j = 2, \dots, m$ and remove all elements of A_1^- . The decrease in value is then bounded by

$$h(\alpha_{i_2} - \alpha_{i_1}) + h(\alpha_{i_3} - \alpha_{i_2}) + \dots + h(\alpha_{i_m} - \alpha_{i_{m-1}}) + \alpha_{A_1^-} \leq h\alpha_{i_m} \leq h.$$

Denote by v_A, v'_A the integral and fractional optimum respectively, with respect to an item list A . Hence,

$$v_A \geq v_{A^-} - h. \tag{4.6}$$

On the other hand, each feasible fractional packing relative to A also yields a feasible packing of A^- if we replace elements of A_j by the corresponding elements of A_j^- , for $j = 1, \dots, m$. Because $\sum_{F \in \mathcal{F}} \alpha_F z_F = \sum_{i=1}^n \sum_{F \ni i} z_F \alpha_i$, the resulting decrease in value is bounded by

$$h(\alpha_{i_2} - \alpha_{i_1}) + h(\alpha_{i_3} - \alpha_{i_2}) + \dots + h(1 - \alpha_{i_m}) \leq h.$$

Thus,

$$v'_A \leq v'_{A^-} + h. \quad (4.7)$$

Let $\text{gap} = \text{gap}_A = v'_A - v_A$ and $\text{gap}_{A^-} = v'_{A^-} - v_{A^-}$. Then inequalities (4.6) and (4.7) imply

$$\text{gap}_A \leq \text{gap}_{A^-} + 2h. \quad (4.8)$$

Now consider the bin packing game relative to A^- . Assume bin sizes are ordered non-increasingly, *i.e.*,

$$B : b_1 \geq b_2 \cdots \geq b_k,$$

We also divide B into $m + 1$ consecutive sublists

$$B = B_1, B_2, \dots, B_m, R'.$$

Let $h' = \lfloor k/m \rfloor$, hence $|B_j| = h'$ for $j = 1, \dots, m$ and $|R'| < h'$. Define the modified lists

$$B^- = B_1^-, B_2^-, \dots, B_m^-, R'$$

by letting $B_j^- = b_{i_j}, \dots, b_{i_j}$, where b_{i_j} is the smallest bin size in B_j .

Denote by v_B, v'_B the integral and fractional optimum corresponding to a bin list B (and item set A^-) respectively. It is straightforward to see that

$$v_B \geq v_{B^-}. \quad (4.9)$$

Indeed, any feasible (integral) packing of B^- is a feasible (integral) packing of B if we simply pack the feasible sets (which are packed to bins) of B_j^- to (the bins of) B_j , for $j = 1, \dots, m$.

On the other hand, each feasible fractional packing relative to B also yields a feasible fractional packing relative to B^- if we pack the feasible sets of B_j to B_{j-1}^- , for $j = 2, \dots, m$ and remove all feasible sets assigned to B_1 . The resulting decrease in value is then bounded by $\alpha_{B_1} \leq h'$.

This shows

$$v'_B \leq v'_{B^-} + h'. \quad (4.10)$$

Let $\text{gap}_B = v'_B - v_B$ and $\text{gap}_{B^-} = v'_{B^-} - v_{B^-}$. Inequalities (4.9) and (4.10) yield

$$\text{gap}_B \leq \text{gap}_{B^-} + h'. \quad (4.11)$$

As gap_B and gap_{B^-} are both defined relative to item set A^- , we may combine (4.8) and (4.11) to yield

$$\text{gap}(N) \leq \text{gap}_{B^-} + 2h + h'. \quad (4.12)$$

Now observe that B^- has at most $m + h'$ different bin sizes and, similarly, A^- contains at most $m + h$ different item sizes. Furthermore, we may assume w.l.o.g. that $k \leq n$, hence $h' \leq h$. Lemma 4.8 implies

$$\text{gap}(N) \leq m + h + 2h + h' \leq m + 4h.$$

Let $m = \epsilon^{-2}$. Then $h \leq \epsilon^2 n \leq \epsilon a_B$ and, correspondingly,

$$\text{gap}(N) \leq \epsilon^{-2} + 4\epsilon a_B.$$

□

The condition $a_B = k\bar{b} \geq \epsilon n$ in the uniform case reads as $k \geq \epsilon n$. Thus, the above lemma finds an upper bound of the integrality gap in case of k is large ($k \geq \epsilon n$). In the non-uniform case, as we have mentioned earlier (cf. Section 4.3.1), the average bin size can be made arbitrarily small by adding small bins. Thus, in the non-uniform case, we need to take the average bin size \bar{b} into account.

4.4.3 Dealing with small items

We first consider items of size strictly larger than ϵ . Observe that a feasible set contains at most $\lceil 1/\epsilon \rceil$ items, hence the total number of items (used by the optimum fractional packing) is bounded (in $O(a_B)$). Applying Lemma 4.9, we obtain the following bound.

Lemma 4.10. *Let $0 < \epsilon < a_1 \leq \dots \leq a_n$. Then $\text{gap}(N) \leq 4\epsilon^{-4} + 2\epsilon^2 a_B$.*

Proof. Recall the optimization problem (4.2) and let $y^* = (y_F^*)_{F \in \mathcal{F}}$ be an optimal solution of the problem. By induction on the number n of items, we may assume that each item i occurs in some feasible set F with $y_F^* \neq 0$. Because each feasible set contains at most $(\epsilon^{-1} - 1)$ items, we obtain the upper bound

$$n \leq |\text{supp}(y^*)| (\epsilon^{-1} - 1)$$

on the number of items.

Note that each item i with $\sum_{F \in \mathcal{F}} \sum_{F \ni i} y_F^* = 1$ contributes more than ϵ to the objective function value. So there can be no more than α_B/ϵ such items i . Hence,

$$|\text{supp}(y^*)| \leq \alpha_B \epsilon^{-1} + k.$$

This shows that $\alpha_B \geq \eta n$ holds with $\eta = \epsilon^2/2$. Therefore, Lemma 4.9 yields the bound

$$\text{gap}(\mathbb{N}) \leq 4\epsilon^{-4} + 2\epsilon^2 \alpha_B.$$

□

We have already observed that small items (*i.e.*, items of size at most $\epsilon \bar{b}$) do not affect the ϵ -balancedness (*cf.* Theorem 4.3). The following result is actually obtained by applying Lemma 4.10 by replacing ϵ with $\epsilon \bar{b}$.

Lemma 4.11. *Let $0 < \epsilon < 1/4$. Then $k \geq 8(\epsilon \bar{b})^{-5}$ implies $\text{gap}(\mathbb{N}) \leq \epsilon \alpha_B$.*

Proof. By induction on $|\mathbb{N}|$. If all items of \mathbb{N} have size $a_i > \epsilon \bar{b}$, then Lemma 4.10 implies

$$\begin{aligned} \text{gap}(\mathbb{N}) &\leq 4(\epsilon \bar{b})^{-4} + 2(\epsilon \bar{b})^2 \alpha_B \leq \epsilon \alpha_B \\ \Leftrightarrow [\epsilon - 2(\epsilon \bar{b})^2] \alpha_B &\geq 4(\epsilon \bar{b})^{-4} \\ \Leftrightarrow [\bar{b}^{-1} - 2(\epsilon \bar{b})] \alpha_B &\geq 4(\epsilon \bar{b})^{-5} \\ \Leftrightarrow k(1 - 2\epsilon \bar{b}^2) &\geq 4(\epsilon \bar{b})^{-5}. \end{aligned}$$

As $\epsilon < 1/4$, the latter follows from the assumed lower bound on k .

If N contains some item $a_i \leq \epsilon \bar{b}$, consider $\tilde{N} = N \setminus \{a_i\}$. By induction, we have $\text{gap}(\tilde{N}) \leq \epsilon a_B$. Let \tilde{v}, \tilde{v}' be the value of an optimum integral and fractional packing for \tilde{N} respectively. If a_i can be placed into any bin "on top of" a corresponding packing of \tilde{v} , then $v(N) \geq \tilde{v} + a_i$ and $v'(N) \leq \tilde{v}' + a_i$ imply $\text{gap}(N) \leq \text{gap}(\tilde{N}) \leq \epsilon a_B$. Otherwise, if a_i does not fit anywhere, then each bin is filled to at least $b_j - a_i$ in the optimum integral solution for \tilde{N} , hence

$$v(N) \geq \tilde{v} \geq \sum_{j=1}^k (b_j - a_i) = a_B - k a_i \geq a_B - \epsilon k \bar{b} = (1 - \epsilon) a_B$$

and, again, $\text{gap}(N) \leq \epsilon a_B$ follows. \square

We seek to prove that $\epsilon\text{-core}(N) \neq \emptyset$ provided the game defined by N is "large" enough. In the uniform case, a sufficient condition in terms of a lower bound $k = \Omega(\epsilon^{-5})$ was given (cf. [28]). Note, however, that we cannot expect such a result to hold for the non-uniform case. Indeed, consider a fixed instance N_0 with minimal taxation rate ϵ_{N_0} . Adding arbitrarily many small bins (smaller than a_{\min} , the minimum item size), we find that $k \rightarrow \infty$ (as well as $a_B \rightarrow \infty$), while ϵ_{N_0} remain unaffected. The same argument shows that even the assumptions in Lemma 4.11 cannot guarantee ϵ -balancedness.

Thus, it seems that we should restrict our attention to irreducible games. Alternatively, given an arbitrary game N , we first apply the simple packing algorithm to split N into a reduced game N^{red} and a (possibly empty) trivial game N^{triv} . Then, if the reduced part is (still) large, a lower bound on the minimum taxation rate for N_{red} (and hence for N) follows:

Theorem 4.12. *Let $0 < \epsilon < 1/2$ with $\epsilon^{-1} \in \mathbb{N}$. If N is reduced (in particular, if N is irreducible), then $k \geq 2^8(\epsilon \bar{b})^{-5}$ implies $\epsilon\text{-core}(N) \neq \emptyset$.*

Proof. Straightforward: As $k \geq 8(\frac{\epsilon}{2}\bar{b})^{-5}$, we get $\text{gap}(N) \leq \frac{\epsilon}{2} a_B$ from Theorem 4.11 and since N is reduced, greedy packing yields $v(N) \geq \frac{1}{2} a_B$. Hence

$$\epsilon_N = \frac{\text{gap}(N)}{v'(N)} \leq \frac{\text{gap}(N)}{v(N)} \leq \epsilon.$$

□

Thus, roughly speaking, games with empty ϵ -core are either “small” or arise from small games by trivial extensions.

4.5 REMARKS AND OPEN PROBLEMS

Our results reveal a certain tradeoff between the taxation rate ϵ and the average bin size \bar{b} . This is most evident in Theorem 4.12, but also applies elsewhere. For example, the condition $a_i > 1/3$ in Proposition 4.7 could equally be replaced by $\bar{b} \geq 4/5$, since for $\epsilon = 5/12$, we have $\bar{b} \geq 4/5 \Leftrightarrow \epsilon \bar{b} \geq 1/3$ and hence the result can be obtained via Lemma 4.3. It is not clear to us whether this phenomenon is inherent to the non-uniform case. In particular, if we consider

$$\epsilon^* := \inf_{\mathbb{N}} \{\epsilon \mid \epsilon\text{-core}(\mathbb{N}) \neq \emptyset\}$$

where the infimum is taken over all instances of the uniform bin packing game, then it is clear (from [28]) that it suffices to consider only games up to a certain fixed size of $|\mathbb{N}|$. Is this no longer true in the non-uniform case?

Recall the conjecture due to G. J. Woeginger in Chapter 3 that, for the uniform bin packing game, the gap is bounded by a universal constant. Are there any counterexamples in the non-uniform case?

Finally, of course a natural question to ask is whether one can improve upon Theorem 4.2 (saying that $\epsilon^* \leq 1/2$ in the non-uniform case). It is also unclear whether one can improve the bound $5/12$ in Theorem 4.7 for large items $a_i > 1/3$ for all i .

Facility location problem

Location planning concerns problems with opening a subset of feasible facilities (service providers) and assigning each customer (who needs the service) to one of the opened facilities while minimizing the total cost. The total cost consists of opening costs and connection costs.

Location planning has a lot of applications in practice like in areas as energy distribution, oil transportation, medical care and network design etc. In this chapter, we consider the so-called (metric) facility location problem and investigate its approximation algorithms. In the introductory section below, we introduce notations of the facility location problem and briefly review some known results. In Section 5.2, we introduce the factor-revealing LP approach proposed by Jain *et al.* ([36, 37, 57]), which is an LP approach for estimating the approximation ratio of an approximation algorithm. The analysis of the approximation ratio will be discussed in Section 5.3. In Section 5.4, we relate this problem with its corresponding game theoretical version (cooperative facility location game) and obtain some results on the ϵ -core allocation. Finally, in Section 5.6, we remark on some open problems.

5.1 INTRODUCTION

An instance of facility location is defined by a set \mathcal{C} of *cities* and a set \mathcal{F} of *facilities*. In addition, we are given *opening costs* f_i for each facility

$i \in \mathcal{F}$ as well as *connection costs* c_{ij} between facility $i \in \mathcal{F}$ and city $j \in \mathcal{C}$. The task is to open a subset of the facilities and to assign each city j to one of the open facilities i (at a cost of c_{ij}) in such a way that the total cost, *i.e.*, the total opening cost plus the total connection cost is minimized. We consider the metric variant of the problem, where the connection costs satisfy the triangle inequality.

The facility location problem is NP-hard and has received a lot of attention in the literature. Thirty years ago, the first (greedy-based) approximation algorithm due to Hochbaum [35] achieved a ratio of $O(\log n)$ in the general (non-metric) case. In the metric case, the first constant factor 3.16-approximation algorithm was found by Shmoys, Tardos and Aardal [69]. Later, the factor was improved to 2.41 by Guha and Khuller [34], where a greedy improvement algorithm was introduced to balance the tradeoff between the upper bound of the opening costs and the upper bound of the connection costs. Both algorithms are based on LP rounding. Other LP rounding methods and primal-dual methods can be found in ([16, 39, 12, 71]). Jain, Mahdian and Saberi [37, 36] proposed a new and fast greedy augmentation algorithm (the JMS algorithm), which runs in $O(n^3)$ and achieves an approximation factor 1.61. It had been shown by Mahdian, Ye and Zhang [57] that the JMS algorithm is a $(1.11, 1.78)$ – *bifactor approximation*, *i.e.*, the total cost of the solution is bounded by $1.11 \cdot F^* + 1.78 \cdot C^*$, where F^* and C^* are the total opening and total connection cost of some optimal solution respectively. Combining the JMS algorithm with greedy improvement results in a 1.52 approximation [57]. As shown by Byrka ([6, 7]), a modification of Chudak’s [16] LP rounding algorithm yields a randomized LP rounding algorithm, with a bifactor $(1.68, 1.38)$, resulting in a 1.5 approximation if it is combined with the JMS algorithm. In Byrka’s algorithm, a fixed scaling factor γ for the opening costs was employed so as to optimize the approximation ratio. Later, Li [52] proved that if γ is randomly selected according to some distribution, the approximation ratio can (again, in combination with the JMS algorithm) be improved to 1.488, which is the best result currently known.

Regarding hardness results, Guha and Khuller [34] showed that there is no λ -approximation algorithm for the metric facility location problem with $\lambda < 1.463$ unless $\text{NP} \subseteq \text{DTIME}(n^{\log \log n})$. Jain *et al.* [36] generalized the result to the nonexistence of a (λ_f, λ_c) -bifactor approximation algorithm for $\lambda_c < 1 + 2e^{-\lambda_f}$ unless $\text{NP} \subseteq \text{DTIME}(n^{\log \log n})$.

Since the currently best known approximation algorithm (*cf.* Li [52]) combines the JMS algorithm with a randomized algorithm, a natural idea is to seek for possible improvements of the JMS algorithm. We investigate a variant of the JMS algorithm due to Mahdian *et al.* [57] and present a better analysis to the upper bound of the approximation ratio. The upper bound is obtained in a way inspired by Fernandes *et al.* [29]. In contrast to [29], our upper bound computation follows a purely “primal” approach, resulting in a more direct (and hopefully more intuitively appealing) derivation. Besides that, we detect and patch a small gap in earlier upper bound computations (*cf.* Lemma 5.3 in Section 5.3.1). This approach is indeed similar to the strongly factor revealing LP approach proposed by Mahdian and Yan [56], however, we derive this independently and in a different perspective.

5.2 FACTOR-REVEALING LP APPROACH

In the following, we describe the JMS algorithm, whose worst case performance ratio can be bounded by the optimal objective function value of a so-called factor-revealing LP (with a parameter k tending to infinity). The idea behind such a factor-revealing LP is to find some linear constraints relating the problem data with certain parameters computed by the algorithm and then maximize the approximation ratio subject to these constraints.

5.2.1 *The JMS algorithm*

All details of the JMS algorithm and the analysis of the factor revealing LP can be found in [36, 37, 57]. We include a short description here for convenience of the reader.

The algorithm uses a notion of time. As time proceeds, facilities are opened and cities get connected as described in the following. The algorithm starts at time $t = 0$, with each city being unconnected and each facility closed. Each city $j \in \mathcal{C}$ has a *budget* α_j that is equal to t and increases with t until the city gets connected (as described below). At any point in time t , each city j offers some money from its budget to each closed facility i as follows: If j is unconnected, the offer equals $\max(t - c_{ij}, 0)$. If j is already connected to some open facility i' , then its offer to facility i equals $\max(c_{i'j} - c_{ij}, 0)$ (the amount it would save when switching to facility i). Actions are taken as soon as one of the following events occurs. (Simultaneous events may be processed in arbitrary order.)

- (a) For some closed facility i , the total offer it receives from all cities equals f_i . In this case we open facility i and connect each city j offering a positive amount to facility i . The amount that j had offered to i is now called the *contribution* of j towards i , and j is no longer allowed to decrease its contribution to i .
- (b) For some unconnected city j , and some facility i that is already open, the budget of j equals c_{ij} . In this case, we connect city j to facility i . The contribution of j towards i is zero.

The algorithm proceeds in this way, raising the budgets of yet unconnected cities until all cities get connected. Recall that the budget of city j equals α_j when it gets connected (at time $t = \alpha_j$) and is not raised any further afterwards. Thus, $\sum_{j \in \mathcal{C}} \alpha_j$ is the total expense (opening plus connection cost) in the solution constructed by the algorithm.

5.2.2 Factor-revealing LP

The approximation ratio of the JMS algorithm can be bounded by a so-called *factor-revealing LP*. Note that the total cost found by the JMS algorithm is $\sum_{j \in \mathcal{C}} \alpha_j$, thus, the approximation ratio is an upper bound for $\sum_j \alpha_j / (\sum_i f_i + \sum_{i,j} c_{ij})$. By deriving constraints related to f_i , c_{ij} and possibly other variables, we will obtain a linear program whose optimal solution is an upper bound of the approximation ratio.

More precisely, consider a *star* S consisting of a facility with opening cost f as its center (for convenience, we also denote the facility itself by f if no misunderstanding is possible), and k cities $1, \dots, k$. Let d_j denote the connection cost between facility f and city j , and let α_j (as defined above) denote j 's contribution to the total expenses. Thus, assuming that an optimal solution would assign cities $1, \dots, k$ to f , at a total cost of $f + \sum_{j=1}^k d_j$, the approximation ratio of the JMS algorithm can be estimated by maximizing $\sum_j \alpha_j / (f + \sum_j d_j)$, subject to the following constraints (5.1 – 5.4).

First, let us assume w.l.o.g. that

$$\alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_k. \quad (5.1)$$

Since here and in what follows we will be dealing with only one fixed facility f (w.l.o.g.). We drop the convention that cities are indexed by j and facilities are indexed by i and rather use both i and j to denote cities. Consider any $1 \leq j < i \leq k$. Let $r_{j,i}$ denote the connection cost of city j at time α_i^- (immediately before i gets connected). In case j also gets connected at time α_i , i.e., when $\alpha_j = \alpha_i$, we let $r_{ji} = \alpha_i$. Recall that once city j gets connected (at time α_j), its budget remains constant, so it can never get connected to another facility with a higher connection cost. (By contrast, it has a chance to connect to another facility which will be opened later with a lower connection cost.) Thus, for every j , we must have

$$\alpha_j \geq r_{j,j+1} \geq r_{j,j+2} \geq \dots \geq r_{j,k}. \quad (5.2)$$

The amount that city j offers to facility f at time α_i equals $[r_{j,i} - d_j]_+ = \max(r_{j,i} - d_j, 0)$ in case j is already connected (or connects) at time α_i and it equals $[\alpha_i - d_j]_+ = \max(\alpha_i - d_j, 0)$ (because α_i is the current budget) in case j is not yet connected. The total offer of cities to a facility never exceeds its opening cost. Thus, for all $i = 1, \dots, k$, we have

$$\sum_{j=1}^{i-1} [r_{j,i} - d_j]_+ + \sum_{j=i}^k [\alpha_i - d_j]_+ \leq f. \quad (5.3)$$

Finally, note that at time α_i , city j has an open facility at distance at most $r_{j,i}$. Hence, by the triangle inequality, city i has one at distance at most $r_{j,i} + d_j + d_i$. Thus, for every $1 \leq j < i \leq k$,

$$\alpha_i \leq r_{j,i} + d_i + d_j. \quad (5.4)$$

Now we are ready to formulate the following factor-revealing LP.

$$\begin{aligned} & \text{maximize } \sum_{j=1}^k \alpha_j & (5.5) \\ & \text{subject to } f + \sum_{j=1}^k d_j = 1 \\ & \alpha_i \leq \alpha_{i+1}, \quad \forall 1 \leq i < k \\ & r_{j,i} \geq r_{j,i+1}, \quad \forall 1 \leq j < i < k \\ & \alpha_j \geq r_{j,j+1}, \quad \forall 1 \leq j < k \\ & \alpha_i \leq r_{j,i} + d_i + d_j, \quad \forall 1 \leq j < i \leq k \\ & \sum_{j=1}^{i-1} [r_{j,i} - d_j]_+ + \sum_{j=i}^k [\alpha_i - d_j]_+ \leq f, \quad \forall 1 \leq i \leq k \\ & \alpha_j, d_j, f, r_{j,i} \geq 0, \quad \forall 1 \leq j \leq i \leq k. \end{aligned}$$

Computational results seem to indicate that the inequalities $\alpha_j \geq r_{j,j+1}$ have no influence on the maximum value. For this reason, they are ignored in previous descriptions of the factor revealing LP ([37])

and [36]). Yet it turns out that we need them in our proof of the upper bound, more precisely in proving Lemma 5.3 below. This is why we include them here.

Let z_k be the optimal objective value of the factor-revealing LP (5.5). Then $\gamma := \sup_k \{z_k\}$ obviously is an upper bound on the approximation ratio of the JMS algorithm. The following bound was obtained by Jain *et al.*

Theorem 5.1 ([36]). $1.598 \leq \gamma \leq 1.61$.

To analyze the factor-revealing LP and derive an upper bound on the optimal objective function value for all values of parameters, *e.g.* k in (5.5), is often hard. In the analysis of Jain *et al.* [36], some constraints which do not drastically change the optimal objective value are relaxed first. This step can be verified by computers. After simplifying the factor-revealing LP, they find an upper bound on its solution by finding a feasible solution for its dual for every k . Again, this dual solution is found with aid of computational experiments. However, this technique does not guarantee tightness of the analysis, because the multiplying parameters used in the proof are explicitly defined according to experimental results, which are too weak to guarantee a tight upper bound. Inspired from this process, Fernandes *et al.* [29] proposed a better way of analyzing the factor-revealing LP, which leads to a tight upper bound (*cf.* Section 5.3.2 for all details).

We improve the approach of Fernandes *et al.* by giving a more direct and intuitive method for analyzing the factor-revealing LP (*cf.* Section 5.3.1). We will not present the analysis for (5.5) with our method. Instead, we consider a modified JMS algorithm due to Mahdian *et al.* [57] (where upper bound of the approximation ratio is shown indirectly, *cf.* the remark before Section 5.3.2). Thus, our analysis to the modified JMS algorithm is new and probably also has applications to variants of facility location problems (*cf.* Section 5.5).

5.2.3 Scaling and augmentation

Now we investigate an improved version of the JMS algorithm due to Mahdian, Ye and Zhang [57]), which consists of two phases. In the first phase, facility i is opened only when the total offer it receives equals $\delta \cdot f_i$ for some fixed $\delta > 1$. Intuitively, as opening costs are scaled up by a factor δ , facilities receive contributions from more cities. In other words, δ can be interpreted as a (opening) delay factor allowing a tradeoff between connection costs and opening costs.

In the second phase, the opening costs are decreased back to their original values at the same rate, *i.e.*, the scaling factor is decreased from δ to 1. During this process, facilities are opened as soon as their opening cost is fully compensated by “switching gains”. This process is also known to be equivalent with the greedy augmentation procedure introduced by Guha and Khuller [34] (also *cf.* [12]). In the second phase, the scaling factor δ is decreased discretely in L steps for some constant L . Let δ_l denote the value of the scaling factor in the l -th step, which will be fixed later. Thus,

$$\delta = \delta_1 > \delta_2 > \dots > \delta_L = 1.$$

In the following we derive the corresponding factor revealing LP. Let $r_{j,k+l}$ denote the connection cost that city j pays after we change the scaling factor to δ_l and process all facilities as described above. Thus, $r_{j,k+1}$ is the connection cost of city j after the first phase. For any step $l = 1, 2, \dots, L$, if f is not open, the total cost that can be saved by switching to f cannot exceed $\delta_l f$. Thus,

$$\sum_{j=1}^k [r_{j,k+l} - d_j]_+ \leq \delta_l f.$$

If f is already open, then $r_{j,k+l} \leq d_j$ for all j and the inequality also holds.

Now we compute the share of city j in the total cost of the solution found. In the first phase of the algorithm, the share of city

j in the total cost is α_j , consisting of its connection cost $r_{j,k+1}$ and its contribution to the opening cost (with scaling factor δ). Thus, the share of city j contributed to the opening costs in the original instance equals $(\alpha_j - r_{j,k+1})/\delta$. After reducing the scaling factor from δ_l to δ_{l+1} ($l = 1, \dots, L-1$), the connection cost of city j is reduced from $r_{j,k+1}$ to $r_{j,k+l+1}$. Therefore, in this step, the share of city j in the opening cost is $r_{j,k+i} - r_{j,k+i+1}$ with respect to the scaled instance, or $(r_{j,k+i} - r_{j,k+i+1})/\delta_{l+1}$ with respect to the original instance. Thus, when the algorithm terminates, the total share of city j in the facility cost is

$$\frac{\alpha_j - r_{j,k+1}}{\delta} + \frac{\sum_{l=1}^{L-1} r_{j,k+l} - r_{j,k+l+1}}{\delta_{l+1}}.$$

Note that the final connection cost of j is $r_{j,k+L}$. Thus, the share of city j in the total cost is

$$\begin{aligned} & \frac{\alpha_j - r_{j,k+1}}{\delta} + \frac{\sum_{l=1}^{L-1} r_{j,k+l} - r_{j,k+l+1}}{\delta_{l+1}} + r_{j,k+L} \\ &= \frac{\alpha_j}{\delta} + \sum_{l=1}^{L-1} \left(\frac{1}{\delta_{l+1}} - \frac{1}{\delta_l} \right) r_{j,k+l}. \end{aligned}$$

We obtain the following factor revealing LP:

$$\begin{aligned} & \text{maximize } \sum_{j=1}^k \left(\frac{\alpha_j}{\delta} + \sum_{l=1}^{L-1} \left(\frac{1}{\delta_{l+1}} - \frac{1}{\delta_l} \right) r_{j,k+l} \right) & (5.6) \\ & \text{subject to } f + \sum_{j=1}^k d_j = 1 \\ & \alpha_i \leq \alpha_{i+1}, \quad \forall 1 \leq i < k \\ & r_{j,i} \geq r_{j,i+1}, \quad \forall 1 \leq j < i < k \\ & \alpha_j \geq r_{j,j+1}, \quad \forall 1 \leq j < k \\ & \alpha_i \leq r_{j,i} + d_i + d_j, \quad \forall 1 \leq j < i \leq k \\ & \sum_{j=1}^{i-1} [r_{j,i} - d_j]_+ + \sum_{j=i}^k [\alpha_i - d_j]_+ \leq \delta f, \quad \forall 1 \leq i \leq k \end{aligned}$$

$$\sum_{j'=1}^k [r_{j',k+l} - d_{j'}] \leq \delta_l f, \quad \forall 1 \leq l \leq L$$

$$\alpha_j, d_j, f, r_{j,i}, r_{j',k+l} \geq 0,$$

$$\forall 1 \leq j \leq i \leq k, 1 \leq j' \leq k, 1 \leq l \leq L.$$

To model a “continuous decrease” of δ , we need to choose the values of δ_l such that $\delta_l > \delta_{l+1}$ and the limit of $\max_l(\delta_l - \delta_{l+1})$ tends to 0 as L goes to infinity. We let $\delta_l = \delta^{(L-l)/(L-1)}$ and have the following.

Lemma 5.2 ([57]). *Let $w_k(\delta)$ be the optimal objective function value of (5.6) with scaling factor δ and some fixed number L . Let $\gamma(\delta) = \sup_k \{w_k(\delta)\}$. Then the modified JMS algorithm solves the metric facility location problem with approximation ratio $\gamma(\delta)$.*

5.3 ANALYZING THE UPPER BOUND

In the original JMS algorithm (cf. [37, 57]), proving upper bounds of factor revealing LPs are based on experimental observations. In their approach, they explicitly define multiplying parameters for all constraints and obtain an upper bound as a function of these parameters via straightforward calculations. Afterwards, they adjust values of multiplying parameters to prove the best possible upper bound on the approximation ratio. This approach requires a lot of calculations and the obtained upper bound is not always satisfying (cf. [29]). Later, this approach was improved by Fernandes *et al.* [29]. They regard those multiplying parameters as variables and obtain a dual problem of the factor-revealing LP. By constraining the total number of variables to a fixed number, they derive a new LP, called the *upper bound factor-revealing linear program*, which has stronger constraints and a larger optimal objective function value. Thus, the approximation ratio can be obtained by solving this upper bound factor-revealing LP.

In this section, we present a primal approach (inspired by Fernandes *et al.* [29]) for obtaining an upper bound of the factor-revealing LP of the facility location problem. The basic idea is to derive a lin-

ear program, whose optimal solution value is the upper bound of the factor-revealing LP for any parameter k . The primal approach we present in the section below is based on “aggregating” variables. Afterwards, in Section 5.3.2, we also derive the same result using the dual approach of Fernandes *et al.* [29]. It can be seen that the primal approach is more direct and intuitive. Our approach is indeed similar to the strongly factor revealing LP approach proposed by Mahdian and Yan [56], however, we derive this independently and in a different perspective.

5.3.1 Primal approach

For fixed δ , the upper bounds $w_k = w_k(\delta)$ seem to converge as $k \rightarrow \infty$. It is not clear whether the sequence w_k is monotonically increasing, but in any case, in analogy to what has been observed by Jain *et al.* [37], and Fernandes *et al.* [29] in the context of “classical” JMS, it can be shown that

Lemma 5.3. $w_k \leq w_{pk}$, for all $p, k \in \mathbb{N}$.

Before proving this lemma, we mention that the proofs in papers [37, 29] are incomplete (both papers fail to define the new r_{ij} values correctly). For example, Jain *et al.* [37] treats the case $p = 2$ by expanding a given optimal solution $(\alpha_j, d_j, f, r_{j,i})$ of the factor revealing LP (5.5) to a new solution of “dimension” $2k$ by duplicating variables, *i.e.*, setting $\alpha'_{2j-1} = \alpha'_{2j} = \alpha_j$, $r'_{2j-1,2i-1} = r'_{2j-1,2i} = r'_{2j,2i-1} = r'_{2j,2i} = r_{j,i}$, $d'_{2j-1} = d'_{2j} = d_j$ and $f' = 2f$. However, the values of $r'_{2j-1,2j}$ remain undefined. A similar gap in the arguments occurs in the paper of Fernandes *et al.* [29]. We present a formal proof below.

Proof of Lemma 5.3. Let $(\hat{\alpha}_j, \hat{d}_j, \hat{f}, \hat{r}_{j,i}, \hat{r}'_{j',k+l})$ ($1 \leq j < i \leq k$, $1 \leq j' \leq k$, $1 \leq l \leq L$) be an optimal solution of (5.6) with objective function value $w_k = w_k(\delta)$, for some fixed $\delta > 1$. We aim at constructing a feasible solution $(\alpha_j, d_j, f, r_{j,i}, r'_{j',pk+l})$ ($1 \leq j < i \leq pk$, $1 \leq j' \leq pk$, $1 \leq l \leq L$) with corresponding objective function value at least w_k . For $i \in$

$[1, pk]$, we define $\hat{i} := \lceil i/p \rceil$, and define variables $\alpha_j, d_j, f, r_{j,i}, r_{j',pk+l}$ as follows:

$$\alpha_j = \frac{\hat{\alpha}_j}{p}, d_j = \frac{\hat{d}_j}{p}, f = \hat{f},$$

$$r_{j,i} = \begin{cases} \frac{\hat{r}_{j,i}}{p}, & \text{if } j < \hat{i}, \\ \frac{\hat{\alpha}_j}{p}, & \text{if } j = \hat{i}, \end{cases}$$

$$r_{j',pk+l} = \frac{\hat{r}_{j',k+l}}{p}.$$

The first 4 types of constraints of (5.6) are straightforward to verify for these variables. As to the fifth constraint, using the fact that $\hat{\alpha}_j \geq \hat{r}_{j,i}$ and, consequently, $r_{j,i} \geq \hat{r}_{j,i}/p$, we get

$$\alpha_i = \frac{\hat{\alpha}_i}{p} \leq \frac{\hat{r}_{j,i}}{p} + \frac{\hat{d}_i}{p} + \frac{\hat{d}_j}{p} \leq r_{j,i} + d_i + d_j.$$

Also the second last constraints in (5.6), which we refer to as “opening cost constraints”, are satisfied:

$$\begin{aligned} & \sum_{j=1}^{i-1} [r_{j,i} - d_j]_+ + \sum_{j=i}^{pk} [\alpha_i - d_j]_+ \\ &= \sum_{j=1}^{p(\hat{i}-1)} [r_{j,i} - d_j]_+ + \sum_{j=p(\hat{i}-1)+1}^{i-1} [r_{j,i} - d_j]_+ + \sum_{j=i}^{pk} [\alpha_i - d_j]_+ \\ &= \sum_{j'=1}^{\hat{i}-1} p \left[\frac{\hat{r}_{j',i}}{p} - \frac{\hat{d}_{j'}}{p} \right]_+ + \sum_{j=p(\hat{i}-1)+1}^{i-1} \left[\frac{\hat{\alpha}_i}{p} - d_j \right]_+ + \sum_{j=i}^{pk} [\alpha_i - d_j]_+ \\ &= \sum_{j'=1}^{\hat{i}-1} [\hat{r}_{j',i} - \hat{d}_{j'}]_+ + \sum_{j=p(\hat{i}-1)+1}^{pk} [\alpha_i - d_j]_+ \\ &= \sum_{j'=1}^{\hat{i}-1} [\hat{r}_{j',i} - \hat{d}_{j'}]_+ + \sum_{j'=\hat{i}}^{\hat{k}} [\hat{\alpha}_i - \hat{d}_{j'}]_+ \\ &\leq \delta \hat{f} = \delta f. \end{aligned}$$

Similar to the above computation, the last set of constraints are satisfied:

$$\sum_{j'=1}^{pk} [r_{j',pk+l} - d_{j'}]_+ = \sum_{j''=1}^k p \left[\frac{\hat{f}_{j'',k+l}}{p} - \frac{\hat{d}_{j''}}{p} \right]_+ \leq \delta_l f.$$

Finally, the objective function reads

$$\begin{aligned} & \sum_{j=1}^{pk} \left(\frac{\alpha_j}{\delta} + \sum_{l=1}^{L-1} \left(\frac{1}{\delta_{l+1}} - \frac{1}{\delta_l} \right) r_{j,pk+l} \right) \\ &= \sum_{j'=1}^k p \left(\frac{1}{\delta} \frac{\hat{\alpha}_{j'}}{p} + \sum_{l=1}^{L-1} \left(\frac{1}{\delta_{l+1}} - \frac{1}{\delta_l} \right) \frac{\hat{f}_{j',k+l}}{p} \right) \\ &= \sum_{j'=1}^k \left(\frac{\hat{\alpha}_{j'}}{\delta} + \sum_{l=1}^{L-1} \left(\frac{1}{\delta_{l+1}} - \frac{1}{\delta_l} \right) \hat{f}_{j',k+l} \right). \end{aligned}$$

□

Intuitively, the reason why w_k changes only little for high values of k is that among a large number k of cities, there must be many with the same characteristics (*i.e.*, distances from facility locations). As a consequence, w_k for large values of k can be estimated by suitably defined values $\hat{w}_{\hat{k}}$, obtained by solving linear programs similar to (5.6) for much smaller values $\hat{k} \leq k$, obtained from (5.6) by “aggregating variables” as in Theorem 5.4 below. This is, roughly, the “primal” idea behind the approach presented by Fernandes *et al.* [29]:

Theorem 5.4. *Let \hat{k} be a positive integer. Then*

$$\hat{w}_{\hat{k}} := \hat{w}_{\hat{k}}(\delta) := \max \sum_{t=1}^{\hat{k}} \left(\frac{\alpha_t}{\delta} + \sum_{l=1}^{L-1} \left(\frac{1}{\delta_{l+1}} - \frac{1}{\delta_l} \right) r_{t,\hat{k}+l} \right) \quad (5.7)$$

$$\text{s.t. } f + \sum_{s=1}^{\hat{k}} d_s = 1$$

$$\alpha_s \leq \alpha_{s+1}, \quad \forall 1 \leq s < \hat{k}$$

$$r_{t,s} \geq r_{t,s+1}, \quad \forall 1 \leq t < s < \hat{k}$$

$$\alpha_s \geq r_{s,s+1}, \quad \forall 1 \leq s < \hat{k}$$

$$\begin{aligned} \alpha_s &\leq r_{t,s} + d_s + d_t, \quad \forall 1 \leq t < s \leq \hat{k} \\ \sum_{t=1}^{s-1} [r_{t,s} - d_t]_+ + \sum_{t=s+1}^{\hat{k}} [\alpha_s - d_t]_+ &\leq g \cdot f, \quad \forall 1 \leq s \leq \hat{k} \\ \sum_{t'=1}^{\hat{k}} [r_{t',\hat{k}+l} - d_{t'}]_+ &\leq \delta_l f, \quad \forall 1 \leq l \leq L, \\ \alpha_s, d_s, f, r_{t,s}, r_{t',\hat{k}+l} &\geq 0, \quad \forall 1 \leq t \leq s \leq \hat{k}, 1 \leq t' \leq \hat{k}, 1 \leq l \leq L \end{aligned}$$

provides an upper bound on $\sup_p w_p(\delta)$, for $p, L \in \mathbb{N}$.

Remark: Note that (5.7) is only a slight relaxation of (5.6), obtained by dropping the term $[\alpha_s - d_t]_+$ for $t = s$ in the “opening cost inequalities”.

Proof. By Lemma (5.3), it suffices to show $\hat{w}_{\hat{k}} \geq w_{\hat{k}p}$ for $p \in \mathbb{N}$. Let $\alpha_i, d_i, r_{j,i}, 1 \leq j < i \leq \hat{k}p$ and $r_{j',k+l}, 1 \leq j' \leq k$ be an optimal solution of (5.6) for $k = \hat{k}p$. We aggregate sets of p variables into one by setting

$$\begin{aligned} I_s &:= \{(s-1)p + 1, \dots, sp\}, \quad s = 1, \dots, \hat{k}, \\ \hat{\alpha}_s &:= \sum_{i \in I_s} \alpha_i, \quad \hat{d}_s := \sum_{i \in I_s} d_i, \\ \hat{r}_{t,s} &:= \frac{1}{p} \sum_{j \in I_t, i \in I_s} r_{j,i} = \frac{1}{p} \sum_{j=1}^p \sum_{i=1}^p r_{(t-1)p+j, (s-1)p+i}, \quad 1 \leq t < s \leq \hat{k}, \\ \hat{r}_{t',\hat{k}+l} &:= \sum_{j' \in I_{t'}} r_{j',\hat{k}p+l}, \quad 1 \leq t' \leq \hat{k}, 1 \leq l \leq L. \end{aligned}$$

We aim at showing that $(\hat{\alpha}_s, \hat{d}_s, \hat{r}_{t,s}, \hat{r}_{t',\hat{k}+l})$ is feasible for (5.7) and its objective value is at least $w_{\hat{k}}$. First note that

$$\sum_{j=1}^k \alpha_j = \sum_{t=1}^{\hat{k}} \hat{\alpha}_t, \quad \sum_{j=1}^k d_j = \sum_{t=1}^{\hat{k}} \hat{d}_t. \tag{5.8}$$

By (5.8), the objective function of (5.6) and (5.7) are related as follows:

$$w_k = \sum_{j=1}^k \left(\frac{\alpha_j}{\delta} + \sum_{l=1}^{L-1} \left(\frac{1}{\delta_{l+1}} - \frac{1}{\delta_l} \right) r_{j,k+l} \right)$$

$$\begin{aligned}
 &= \sum_{t=1}^{\hat{k}} \frac{\hat{\alpha}_t}{\delta} + \sum_{t=1}^{\hat{k}} \sum_{j=1}^p \sum_{l=1}^{L-1} \left(\frac{1}{\delta_{l+1}} - \frac{1}{\delta_l} \right) r_{(t-1)p+j, t\hat{k}+l} \\
 &= \sum_{t=1}^{\hat{k}} \frac{\hat{\alpha}_t}{\delta} + \sum_{t=1}^{\hat{k}} \sum_{l=1}^{L-1} \left(\frac{1}{\delta_{l+1}} - \frac{1}{\delta_l} \right) \hat{r}_{t, \hat{k}+l} \\
 &\leq \hat{w}_{\hat{k}},
 \end{aligned}$$

provided that we can show that $\hat{\alpha}_s$, \hat{d}_s , $\hat{r}_{t,s}$ and $\hat{r}_{t, \hat{k}+l}$ are feasible for (5.7). This is straightforward to verify: The first constraint follows immediately from (5.8) and the following two sets of constraints are trivially satisfied. As to the third, observe that for $j \in I_s$ and $i \in I_{s+1}$, $\alpha_j \geq r_{j,i}$. Thus $\alpha_j \geq \frac{1}{p} \sum_{i \in I_s} r_{j,i}$ and hence $\sum_{j \in I_s} \alpha_j \geq \frac{1}{p} \sum_{j \in I_s} \sum_{i \in I_{s+1}} r_{j,i}$, i.e., $\hat{\alpha}_s \geq \hat{r}_{s,s+1}$ is fulfilled.

As to the fourth set of constraints (the “triangle inequality constraints”), note that for $1 \leq t < s \leq \hat{k}$ we have

$$\begin{aligned}
 p\hat{\alpha}_s &= \sum_{j=1}^p \sum_{i=1}^p \alpha_{(s-1)p+i} \\
 &\leq \sum_{j=1}^p \sum_{i=1}^p r_{(t-1)p+j, (s-1)p+i} + \sum_{j=1}^p \sum_{i=1}^p d_{(s-1)p+i} \\
 &\quad + \sum_{j=1}^p \sum_{i=1}^p d_{(t-1)p+i} \\
 &\leq p\hat{r}_{t,s} + p\hat{d}_s + p\hat{d}_t,
 \end{aligned}$$

yielding the desired inequality.

Now consider the “opening cost” constraints. Summing up the original opening cost constraints in (5.6) for $i = (s-1)p+1, \dots, sp$ yields

$$\begin{aligned}
 p \cdot \delta f &\geq \sum_{i=(s-1)p+1}^{sp} \sum_{j=1}^{i-1} [r_{j,i} - d_j]_+ + \sum_{i=(s-1)p+1}^{sp} \sum_{j=i}^{\hat{k}p} [\alpha_i - d_j]_+ \\
 &\geq \sum_{i=(s-1)p+1}^{sp} \sum_{j=1}^{(s-1)p} [r_{j,i} - d_j]_+ + \sum_{i=(s-1)p+1}^{sp} \sum_{j=sp+1}^{\hat{k}p} [\alpha_i - d_j]_+
 \end{aligned}$$

$$\begin{aligned}
 &\geq \sum_{t=1}^{s-1} \sum_{j=1}^p \sum_{i=1}^p [r_{(t-1)p+j, (s-1)p+i} - d_{(t-1)p+j}]_+ \\
 &\quad + \sum_{t=s+1}^{\hat{k}} \sum_{j=1}^p \sum_{i=1}^p [\alpha_{(s-1)p+i} - d_{(t-1)p+j}]_+ \\
 &\geq p \cdot \sum_{t=1}^{s-1} [\hat{r}_{t,s} - \hat{d}_t]_+ + p \cdot \sum_{t=s+1}^{\hat{k}} [\hat{\alpha}_s - \hat{d}_t]_+,
 \end{aligned}$$

where the last inequality holds as $[a]_+ + [b]_+ \geq [a + b]_+$. Hence, indeed also the opening cost constraints

$$\delta f \geq \sum_{t=1}^{s-1} [\hat{r}_{t,s} - \hat{d}_t]_+ + \sum_{t=s+1}^{\hat{k}} [\hat{\alpha}_s - \hat{d}_t]_+.$$

in (5.7) are satisfied.

Finally, the last set of constraints related to the augmentation process read

$$\begin{aligned}
 \delta_1 f &\geq \sum_{j'=1}^{\hat{k}_p} [r_{j', \hat{k}_{p+1}} - d_{j'}]_+ \\
 &= \sum_{t'=1}^{\hat{k}} \sum_{j'=1}^p [r_{(t'-1)p+j', \hat{k}_{p+1}} - d_{(t'-1)p+j'}]_+ \\
 &= \sum_{t'=1}^{\hat{k}} [\hat{r}_{t', \hat{k}_{p+1}} - d_{t'}]_+
 \end{aligned}$$

and the proof is complete. \square

Solving (5.7) with $\delta = 1.504$, $\hat{k} = 400$, $L = 100$ by CPLEX, we get $\hat{w}_{400}(1.504) = 1.5180$, which is quite close to the lower bound. Indeed, a similar computation yields $w_{400}(1.504) = 1.5161$ as a lower bound for the true approximation ratio.

Remark. Mahdian *et al.* [57] proved the above approximation ratio in a somewhat more circumstantial way. It is known that the process of decreasing δ to 1 is equivalent to the greedy augmenting procedure

due to Guha and Khuller [34], where it is also stated that a bifactor (γ_f, γ_c) approximation algorithm implies a $(\gamma_f + \ln \delta + \epsilon, 1 + \frac{\gamma_c - 1}{\delta})$ approximation algorithm via greedy augmenting. It is also known that the JMS algorithm is a $(1.11, 1.78)$ approximation algorithm (cf. [57]). Taking $\delta = 1.504$ gives a 1.52-approximation algorithm.

5.3.2 Dual approach

In this section, we compare our derivation from Section 5.3.1 with the dual approach by Fernandes *et al.* [29]. The result will be the LP-dual of (5.7). The general idea works as follows: First consider the dual program of the factor-revealing LP. By strong duality theorem, the optimal objective function value must be equal to that of the factor-revealing LP. Thus, the upper bound of the dual program also gives the approximation factor for our facility location problem. In general, computing the upper bound for the dual program is difficult (as k might be arbitrarily large). Therefore, we relax this program by constraining the total number of variables to a fixed number t , which is independent of k . By Lemma 5.3, we can assume that k has the form $k = p \cdot t$, where p, t are positive integers. Thus, for any fixed t , the number p can be regarded as a scaling factor, which can be used to scale the total number of variables in some new linear program. Since any solution of the new linear program yields an upper bound on the approximation factor, it is called an *upper bound factor-revealing linear program* (cf. [29] for more details).

First, the factor-revealing linear program (5.6) can be equivalently written as

$$\begin{aligned} & \text{maximize} \quad \sum_{j=1}^k \left(\frac{\alpha_j}{\delta} + \sum_{l=1}^{L-1} \left(\frac{1}{\delta_{l+1}} - \frac{1}{\delta_l} \right) r_{j,k+l} \right) & (5.9) \\ & \text{subject to} \quad f + \sum_{j=1}^k d_j \leq 1 \\ & \quad \alpha_i \leq \alpha_{i+1}, \quad \forall 1 \leq i < k \end{aligned}$$

$$\begin{aligned}
 & r_{j,i} \geq r_{j,i+1}, \quad \forall 1 \leq j < i < k \\
 & \alpha_i \leq r_{j,i} + d_i + d_j, \quad \forall 1 \leq j < i \leq k \\
 & r_{j,i} - d_j \leq x_{j,i}, \quad \forall 1 \leq j < i \leq k \\
 & \alpha_i - d_j \leq x_{j,i}, \quad \forall 1 \leq i \leq j \leq k \\
 & \sum_{j=1}^k x_{j,i} \leq \delta f, \quad \forall 1 \leq i \leq k \\
 & r_{j',k+l} - d_{j'} \leq x_{j',k+l}, \quad \forall 1 \leq j' \leq k, 1 \leq l \leq L \\
 & \sum_{j'=1}^k x_{j',k+l} \leq \delta_l f, \quad \forall 1 \leq l \leq L \\
 & \alpha_j, d_j, f, r_{j,i}, r_{j',k+l}, x_{j,i}, x_{j',k+l} \geq 0, \\
 & \forall 1 \leq j \leq i \leq k, 1 \leq j' \leq k, 1 \leq l \leq L.
 \end{aligned}$$

We introduce variables $\gamma, \alpha_i, b_{j,i}, c_{j,i}, e_{j,i}, g_i, e_{j,k+l}, h_l$ corresponding to constraints of (5.9) and obtain its dual linear program as below.

minimize γ (5.10)

$$\begin{aligned}
 \text{subject to } & \alpha_i - \alpha_{i-1} + \sum_{j=1}^{i-1} c_{j,i} + \sum_{j=i}^k e_{j,i} \geq \frac{1}{\delta}, \quad \forall 1 \leq i \leq k \\
 & b_{j,i-1} - b_{j,i} - c_{j,i} + e_{j,i} \geq 0, \quad \forall 1 \leq j < i \leq k \\
 & \gamma - \sum_{j=1}^{i-1} c_{j,i} - \sum_{j=i+1}^k c_{i,j} - \sum_{j=1}^k e_{i,j} - \sum_{l=1}^L e_{i,k+l} \geq 0, \\
 & \forall 1 \leq i \leq k \\
 & \gamma - \delta \sum_{i=1}^k g_i - \sum_{l=1}^L h_l \delta_l \geq 0 \\
 & g_i - e_{j,i} \geq 0, \quad \forall 1 \leq i, j \leq k \\
 & e_{j,k+l} \geq \frac{1}{\delta_{l+1}} - \frac{1}{\delta_l}, \quad \forall 1 \leq j \leq k, 1 \leq l \leq L-1 \\
 & h_l - e_{j,k+l} \geq 0, \quad \forall 1 \leq j \leq k, 1 \leq l \leq L \\
 & \alpha_0 = \alpha_k = b_{i,i} = b_{i,k} = 0, \quad \forall 1 \leq i \leq k \\
 & \alpha_i, b_{j,i}, c_{j,i}, e_{j,i}, g_i, e_{j,k+l}, h_i \geq 0,
 \end{aligned}$$

$$\forall 1 \leq j \leq i \leq k, 1 \leq l \leq L.$$

By the strong duality theorem, the optimal objective function value of (5.10) equals that of (5.9). Next, we aim at finding an upper bound LP of (5.10) for any k . The idea is to restrict the variables to a small program to obtain a feasible solution for the dual program. To obtain a linear program independent of k , we will scale the variables by p . W.l.o.g., assume k has the form $k = pt$, with p, t positive integers. Let $\hat{n} := \lceil n/p \rceil$, for any number n and define variables $\gamma', a'_i, b'_{j,i}, c'_{j,i}, e'_{j,i}, g'_i, e'_{j,k+l}, h'_l$ as below.

$$\gamma = \gamma', \quad (5.11)$$

$$a_i = pa'_i - (p\hat{i} - i)(a'_i - a'_{i-1}), \quad (5.12)$$

$$b_{j,i} = b'_{j,\hat{i}} - \frac{p\hat{i} - i}{p}(b'_{j,\hat{i}} - b'_{j,\hat{i}-1}), \quad (5.13)$$

$$c_{j,i} = \frac{c'_{j,\hat{i}}}{p}, \quad (5.14)$$

$$e_{j,i} = \frac{e'_{j,\hat{i}}}{p}, \quad (5.15)$$

$$g_i = \frac{g'_i}{p}, \quad (5.16)$$

$$e_{j,k+l} = e'_{j,k+l}, \quad (5.17)$$

$$h_l = h'_l, \quad (5.18)$$

$$a'_0 = a'_t = b_{i,\hat{i}} = b_{i,t} = 0. \quad (5.19)$$

Notice that $a_i - a_{i-1} = a'_i - a'_{i-1}$ and that $b_{j,i-1} - b_{j,i} = (b'_{j,\hat{i}-1} - b'_{j,\hat{i}})/p$. To see this, it is enough to use the fact that

$$\widehat{i-1} = \begin{cases} \hat{i} - 1, & \text{if } i \equiv 1 \pmod{p}, \\ \hat{i}, & \text{otherwise,} \end{cases}$$

and to distinguish these two cases.

In the following, we apply definitions (5.11 – 5.19) and derive a set of “stronger constraints” for the new variables γ', a'_i etc. By definitions (5.11 – 5.19), we conclude that

$$\begin{aligned} & a_i - a_{i-1} + \sum_{j=1}^{i-1} c_{j,i} + \sum_{j=1}^k e_{j,i} \\ \geq & a'_i - a'_{i-1} + \sum_{j=1}^{i-1} \frac{c'_{j,\hat{i}}}{p} + \sum_{j=i}^{pt} \frac{e'_{j,\hat{i}}}{p} \\ \geq & a'_i - a'_{i-1} + \sum_{j'=1}^{\hat{i}-1} p \frac{c'_{j',\hat{i}}}{p} + \sum_{j'=\hat{i}+1}^t p \frac{e'_{j',\hat{i}}}{p} \\ = & a'_i - a'_{i-1} + \sum_{j'=1}^{\hat{i}-1} c'_{j',\hat{i}} + \sum_{j'=\hat{i}+1}^t e'_{j',\hat{i}} \geq \frac{1}{\delta}, \\ & b_{j,i-1} - b_{j,i} - c_{j,i} + e_{j,i} = \frac{b'_{j,\hat{i}-1} - b'_{j,\hat{i}}}{p} + \frac{e'_{j,\hat{i}}}{p} - \frac{c'_{j,\hat{i}}}{p} \geq 0, \end{aligned}$$

$$\begin{aligned} & \gamma - \sum_{j=1}^{i-1} c_{j,i} - \sum_{j=i+1}^k c_{j,i} - \sum_{j=1}^k e_{i,j} - \sum_{l=1}^L e_{i,k+l} \\ = & \gamma' - \sum_{j=1}^{i-1} \frac{c'_{j,\hat{i}}}{p} - \sum_{j=i+1}^k \frac{c'_{j,\hat{i}}}{p} - \sum_{j=1}^k \frac{e'_{i,j}}{p} - \sum_{l=1}^L e_{i,k+l} \\ \geq & \gamma' - \sum_{j'=1}^{\hat{i}} p \frac{c'_{j',\hat{i}}}{p} - \sum_{j'=\hat{i}}^t p \frac{c'_{j',\hat{i}}}{p} - \sum_{j'=1}^t p \frac{e'_{i,j'}}{p} - \sum_{l=1}^L e_{i,k+l} \\ = & \gamma' - \sum_{j'=1}^{\hat{i}-1} c'_{j',\hat{i}} - \sum_{j'=\hat{i}+1}^t c'_{j',\hat{i}} - \sum_{j'=1}^t e'_{i,j'} - \sum_{l=1}^L e_{i,k+l} \geq 0, \end{aligned}$$

$$\begin{aligned} & \gamma - \delta \sum_{i=1}^k g_i - \sum_{l=1}^L h_l \delta_l = \gamma' - \delta \sum_{i=1}^k \frac{g'_i}{p} + \sum_{l=1}^L h'_l \delta_l \\ = & \gamma' - \delta \sum_{i=1}^t p \frac{g'_{i'}}{p} + \sum_{l=1}^L h'_l \delta_l = \gamma' - \delta \sum_{i'=1}^t g'_{i'} + \sum_{l=1}^L h'_l \delta_l \geq 0. \end{aligned}$$

The remaining constraints are trivially satisfied due to definitions (5.11 – 5.19). Thus, we arrive at a linear program as below:

$$\begin{aligned}
& \text{minimize } \gamma & (5.20) \\
& \text{subject to } a_i - a_{i-1} + \sum_{j=1}^{i-1} c_{j,i} + \sum_{j=i+1}^t e_{j,i} \geq \frac{1}{\delta}, \quad \forall 1 \leq i \leq t \\
& \quad b_{j,i-1} - b_{j,i} - c_{j,i} + e_{j,i} \geq 0, \quad \forall 1 \leq j < i \leq t \\
& \quad \gamma - \sum_{j=1}^{i-1} c_{j,i} - \sum_{j=i+1}^t c_{i,j} - \sum_{j=1}^t e_{i,j} - \sum_{l=1}^L e_{i,t+l} \geq 0, \\
& \quad \forall 1 \leq i \leq t \\
& \quad \gamma - \delta \sum_{i=1}^t g_i - \sum_{l=1}^L h_l \delta_l \geq 0 \\
& \quad g_i - e_{j,i} \geq 0, \quad \forall 1 \leq i, j \leq t \\
& \quad e_{j,t+l} \geq \frac{1}{\delta_{l+1}} - \frac{1}{\delta_l}, \quad \forall 1 \leq j \leq t, 1 \leq l \leq L-1 \\
& \quad h_l - e_{j,t+l} \geq 0, \quad \forall 1 \leq j \leq t, 1 \leq l \leq L \\
& \quad a_0 = a_t = b_{i,i} = b_{i,t} = 0, \quad \forall 1 \leq i \leq t \\
& \quad a_i, b_{j,i}, c_{j,i}, e_{j,i}, g_i, e_{j,t+l}, h_i \geq 0, \\
& \quad \forall 1 \leq j \leq i \leq t, 1 \leq l \leq L.
\end{aligned}$$

As we have seen above, any feasible solution to (5.10) implies a feasible solution to (5.20). Therefore, (5.20) for arbitrary t yields an upper bound on (5.10). We therefore call (5.20) an upper bound factor-revealing linear program corresponding to (5.10). Furthermore, if we calculate the dual program of (5.20), we obtain (5.7).

5.4 FACILITY LOCATION GAMES

As motivated in Chapter 2, cooperative facility location games concern problems of finding a fair cost allocation for all players (facilities and cities). Thus, assume that the player set N consists of all facilities in \mathcal{F} and all cities in \mathcal{C} . For any coalition $S \subseteq N$ containing facilities and

cities, their total cost $c(S)$ is the minimal cost of the related facility location problem. (If $S \cap \mathcal{F} = \emptyset$, we let $c(S) = \infty$.)

Cooperative facility location games have been investigated by Goemans and Skutella [32]. They showed that testing non-emptiness of the core and testing core membership of an allocation vector for the unconstrained facility location game (*i.e.* the number of cities connected to a facility is unlimited) are both NP-complete. Moreover, they showed that the LP relaxation (*cf.* (5.21) below) is equivalent to the allocation problem (2.4) (*cf.* Chapter 2).

We define binary variables y_i indicating whether a facility $i \in \mathcal{F}$ is opened and x_{ij} indicating whether a city $j \in \mathcal{C}$ is connected to facility i . Thus, the problem can be formulated as the following integer linear program:

$$\begin{aligned} & \text{minimize} && \sum_{i \in \mathcal{F}} f_i y_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{C}} c_{ij} x_{ij} && (5.21) \\ & \text{subject to} && \sum_{i \in \mathcal{F}} x_{ij} = 1, \quad \forall j \in \mathcal{C}, \\ & && x_{ij} \leq y_i, \quad \forall i \in \mathcal{F}, j \in \mathcal{C}, \\ & && x_{ij}, y_i \in \{0, 1\}, \quad \forall i \in \mathcal{F}, j \in \mathcal{C}. \end{aligned}$$

The first set of constraints indicates that each city is connected to exactly one facility. The second set of inequalities assures that cities can only be assigned to open facilities. The relaxation of (5.21) is

$$\begin{aligned} & \text{minimize} && \sum_{i \in \mathcal{F}} f_i y_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{C}} c_{ij} x_{ij} && (5.22) \\ & \text{subject to} && \sum_{i \in \mathcal{F}} x_{ij} = 1, \quad \forall j \in \mathcal{C}, \\ & && x_{ij} \leq y_i, \quad \forall i \in \mathcal{F}, j \in \mathcal{C}, \\ & && x_{ij}, y_i \geq 0, \quad \forall i \in \mathcal{F}, j \in \mathcal{C}. \end{aligned}$$

It can be easily observed that an optimal solution y of (5.21) has $y_i \leq 1$ for all $i \in \mathcal{F}$.

Let $c(N), c'(N)$ be the optimal solution value corresponding to (5.21) and (5.22) respectively. By Lemma 2.4, the non-emptiness of the ϵ -

core for (unconstrained) facility location games can be characterized as below.

Lemma 5.5. $\epsilon\text{-core}(N) \neq \emptyset$ iff $\epsilon \geq c(N)/c'(N) - 1$.

Observe that now $c(N)$ concerns a minimization problem, the above condition becomes $c(N)/c'(N) - 1$, instead of $1 - v(N)/v'(N)$ in case $v(N)$ stands for a maximization problem.

As before, the minimum ϵ which ensures a nonempty ϵ -core (or equivalently, the minimal taxation rate) is determined by the relative integrality gap $c(N)/c'(N)$. In the metric case, Chudak [16] gave an approximation algorithm based on LP rounding and showed that the relative integrality gap $c(N)/c'(N) \leq 1 + 2/e \approx 1.736$, implying that the 0.736 core is nonempty for all N . On the negative side, Guha and Khuller [34] proved a lower bound of 1.463 on the best possible upper bound of the relative integrality gap by a reduction from Set Cover, assuming $\text{NP} \neq \text{DTIME}[n^{O(\log \log n)}]$.

We observe that the improved JMS algorithm (with scaling and augmentation) automatically yields a 0.52-core allocation by allocating α_j to city j (where α_j here denotes the share of city j in the total cost found by the algorithm). Indeed, for any $S \subseteq N$ consisting of facilities and cities, we consider a star from an optimal solution of S , where f is the opening cost and d_1, \dots, d_k are the corresponding connection costs. By the improved JMS algorithm, we have $\sum_{i=1}^k \alpha_i \leq 1.52(f + \sum_{i=1}^k d_i)$. Thus, by Lemma 5.5, this in turn shows that $c(N)/c'(N) \leq 1.52$.

In addition, LP rounding gives us another insight on the (relative) integrality gap. Next, we briefly introduce a randomized LP rounding algorithm due to Byrka [6], which results in a better upper bound on the relative integrality gap if it is combined with the JMS algorithm. Let (x^*, y^*) be an optimal solution of (5.22) and let γ be a scaling factor. The idea is to scale up y^* by γ and using them to fix the x -variables by connecting the cities to their nearest open facilities (but still respecting the constraint $x_{ij} \leq \gamma y_i$). We denote the obtained new solution by (\bar{x}, \bar{y}) . W.l.o.g., we can assume that each city is fully served by frac-

tional facilities, *i.e.*, $\bar{x}_{ij} \in \{0, \bar{y}_i\}$ for all $i \in \mathcal{F}, j \in \mathcal{C}$ (otherwise, we can split facilities if necessary). Clearly, the total opening cost of the new solution is bounded by γF^* , where $F^* = \sum_{i \in \mathcal{F}} f_i y_i^*$ is the total opening cost of the optimal solution of (5.22).

If facilities that fractionally serve a client j are all at the same distance, for such an instance, the algorithm of Chudak and Shmoys [17] leads to a 1.463 approximation algorithm. Therefore, difficult instances have distinct connection costs. For this reason, we define the set of *close* facilities $C(j)$ to be facilities that serve j in (\bar{x}, \bar{y}) . Those facilities that serve j in (x^*, y^*) but not in (\bar{x}, \bar{y}) are called *distant* facilities and are denoted by $D(j)$. Let $d_{av}^C(j)$, $d_{av}^D(j)$ be the average distance from j to close and distant facilities (w.r.t. fractions \bar{y} or equivalently y) respectively. Define $d_{max}^C(j)$ to be the maximum distance from j to a facility in $C(j)$.

Now we consider the *support graph* of an LP solution (x^*, y^*) , which is a bipartite graph, with node set $\mathcal{C} \cup \mathcal{F}$ and an edge between two nodes if a city is fractionally served by a facility. The clustering of this graph is a partitioning of clients into clusters, with an assigned *cluster center* (which is a city) and with a requirement that two cluster centers are not served by the same facility.

The LP rounding algorithm due to Byrka [6] is as follows:

1. Greedily compute a clustering for (\bar{x}, \bar{y}) by setting cluster centers that minimize $d_{av}^C(j) + d_{max}^C(j)$.
2. For every cluster center j , open one of its close facilities randomly with probabilities \bar{x}_{ij} .
3. For each facility i that is not a close facility of any cluster center, open it with probability \bar{y}_i .
4. Connect each city to the closest open facility.

Let C_{sol} be the connection cost found by the above algorithm. Byrka proved the following result:

Theorem 5.6 ([6]). $E(C_{sol}) \leq 1 + 2/e^\gamma$, for $\gamma \geq 1.67736$.

Thus, this is a $(\gamma, 1 + 2/e^\gamma)$ bi-factor approximation algorithm. Besides, a 1.5-approximation algorithm is attained if this algorithm is combined with the JMS algorithm (with $\gamma = 1.67736$). Afterwards, an improvement was made by Li [52], who showed that if γ is randomly chosen according a certain distribution, the approximation ratio can be improved to 1.488 (in expected value) if combined with the JMS algorithm, which also implies the relative integrality gap $c(N)/c'(N) \leq 1.488$. Thus, regarding the approximate core allocation, we claim

Theorem 5.7. *The 0.488-core of the metric facility location game is nonempty.*

5.5 APPLICATION TO VARIANTS OF THE PROBLEM

In this section, we describe the application of the JMS algorithm (not the modified JMS) to several variants of the metric facility location problem. The reader can also refer to Jain *et al.* [36]. Hopefully, the modified JMS also works for these variants hence our analysis to the modified JMS algorithm can be also applied.

5.5.1 The k -median problem

In the k -median problem, we are also given a set of facilities and a set of cities. The difference between k -median and facility location is that there are no opening costs for facilities, while instead, there is a limit k on the total number of facilities to be opened. A generalization of the k -median problem called k -facility location, takes opening costs into account. The k -median problem is also a classical problem in combinatorial optimization and has been intensively studied in the literature ([1, 13, 12, 38, 15]). The best approximation algorithm due to Li and Svensson [53] achieves a factor $1 + \sqrt{3} + \epsilon$. As observed by Jain *et al.* [36], the hardness result of facility location due to Ghua and Khuller [34] can be adapted to show the non-existence of $1 + 2/e - \epsilon \approx 1.736$ approximation algorithm for k -median, un-

less $\text{NP} \subseteq \text{DTIME}(n^{O(\log \log n)})$. Thus, k -median is strictly harder to approximate than facility location as the latter can be approximated within a factor of 1.488.

Approximation algorithms for the facility location problem may be used to solve the k -median problem by assigning some virtual cost z to each facility. If $z = 0$, then the algorithm opens all facilities. For $z \rightarrow \infty$, the algorithm opens exactly one facility, which is a feasible solution to the k -median problem. Thus, a suitable choice of z can be used for approximating the optimal solution of the k -median problem.

Suppose A is an approximation algorithm for the facility location problem. Consider an instance I of the facility location problem with optimal cost OPT . Let F_A and C_A be the opening cost and the connection cost found by A for the instance I . A is called a *Lagrangian Multiplier Preserving* α -approximation algorithm (LMP α -approximation algorithm for short, cf. [38, 36]) if $C_A \leq \alpha(\text{OPT} - F_A)$ for all I . Jain and Vazirani proved the following result.

Lemma 5.8 ([38]). *An LMP α -approximation algorithm for the facility location problem gives a 2α -approximation algorithm for the k -facility location problem.*

As k -facility is a generalization of k -median, the lemma also holds for the k -median problem.

Jain *et al.* [36] showed that an α -approximation algorithm for the metric facility location problem yields an LMP α -approximation algorithm for the same problem: Consider an instance I of the metric facility location problem. Assume A is a $(1, \alpha)$ bi-factor approximation algorithm for some α , *i.e.*,

$$A \leq F^* + \alpha C^*, \tag{5.23}$$

where F^*, C^* are opening cost and connection cost of an optimal solution of I respectively.

We run A on the modified instance I' , where I' is obtained from I by scaling up the opening costs by α and output the solution. In the following we show this yields an LMP α -approximation.

Let $\sum_{j \in C} \alpha_j = F_A + C_A$ be the total cost of the solution for I , where F_A, C_A denote the opening cost and the connection cost respectively. Thus the total cost for I' is $\alpha F_A + C_A$. Since the optimal solution for I is feasible for I' , by (5.23), we have

$$\alpha F_A + C_A \leq \alpha F^* + \alpha C^* = \alpha \text{OPT},$$

implying $C_A \leq \alpha(\text{OPT} - F_A)$. Thus, this is an LMP α -approximation algorithm.

To determine a suitable value of α , it suffices to find an upper bound for the factor-revealing LP (5.5) with the opening costs subtracted. Thus, we obtain the following factor-revealing LP:

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^p \alpha_i - f && (5.24) \\ & \text{subject to} && \sum_{i=1}^p d_i = 1 \\ & && \alpha_i \leq \alpha_{i+1}, \quad \forall 1 \leq i < p \\ & && r_{j,i} \geq r_{j,i+1}, \quad \forall 1 \leq j < i < p \\ & && \alpha_j \geq r_{j,j+1}, \quad \forall 1 \leq j < p \\ & && \alpha_i \leq r_{j,i} + d_i + d_j, \quad \forall 1 \leq j < i \leq p \\ & && \sum_{j=1}^{i-1} [r_{j,i} - d_j]_+ + \sum_{j=i}^p [\alpha_i - d_j]_+ \leq f, \quad \forall 1 \leq i \leq p \\ & && \alpha_j, d_j, f, r_{j,i} \geq 0, \quad \forall 1 \leq j \leq i \leq p. \end{aligned}$$

Solving (5.24) with $p = 100$, we get a lower bound 1.99. Jain *et al.* [36] proved that the upper bound is precisely 2 by presenting the following feasible solution to the corresponding upper bound factor-revealing LP (*cf.* Section 5.3):

$$\alpha_j = \begin{cases} 2 - 1/p, & j = 1 \\ 2, & 2 \leq j \leq p \end{cases}$$

$$d_j = \begin{cases} 1, & j = 1 \\ 0, & 2 \leq j \leq p \end{cases}$$

$$r_{j,i} = \begin{cases} 1, & j = 1 \\ 2, & 2 \leq j \leq p \end{cases}$$

$$f = 2(p - 1).$$

By Lemma 5.8, it implies a 4-approximation algorithm for the k -median problem. (Though the approximation factor is not the currently best, the approach itself is interesting anyway and could be improved.)

5.5.2 Facility location with arbitrary demands

For the facility location with arbitrary demands, there is a nonnegative integer demand μ_j for each city j . An open facility i can serve one unit of demand at a cost c_{ij} . One way to look at this modification is to reduce it to the unit demand case by considering μ_j copies of city j instead. This reduction suggests to modify our algorithm such that city j raises its contribution at rate μ_j . Thus, the algorithm still achieves the same approximation ratio. (In case μ_j 's are fractional numbers, we can scale them to integers. Thus, the modified algorithm works in general as well.)

We remark that we have no idea about how to obtain a corresponding factor-revealing LP for this case. Indeed, let t_1, t_2, \dots, t_k be the sequence of times at which cities get connected. At any moment t_i when i gets connected, its budget is $\alpha_i = t_i \mu_i$. Similar to (5.1), we can assume $t_1 \leq t_2 \leq \dots \leq t_k$, *i.e.*,

$$\frac{\alpha_1}{\mu_1} \leq \frac{\alpha_2}{\mu_2} \leq \dots \leq \frac{\alpha_k}{\mu_k}.$$

The inequalities (5.2) $r_{j,i} \geq r_{j,i+1}$ for $1 \leq j < i < k$ still hold as before. As to the opening cost inequality (5.3), the amount that city i offers to facility f at time t_i equals $[r_{j,i} \mu_j - d_j \mu_j]_+$ in case j is already

connected (or connects) at time t_i and it equals $[\alpha_i - d_j \mu_j]_+$ (because α_i is the current budget) in case j is not yet connected. As before, the total offer of cities to a facility never exceeds its opening cost. Thus, for $i = 1, 2, \dots, k$, the opening cost inequality becomes

$$\sum_{j=1}^{i-1} [r_{j,i} \mu_j - d_j \mu_j]_+ + \sum_{j=i}^k [\alpha_i - d_j \mu_j]_+ \leq f.$$

Finally, it is not hard to derive the triangle inequality: for $1 \leq j < i \leq k$,

$$\frac{\alpha_i}{\mu_i} \leq r_{j,i} + d_i + d_j.$$

Obviously, the inequalities above are non-linear. Thus, it is rather unclear whether we can derive a corresponding factor-revealing LP as before.

5.5.3 Facility location with penalties

Assume that some cities are far away from facility locations. Connecting these cities to facilities may have strong influence on the optimum cost. Another way of dealing with this problem is to consider these distant cities as *outliers*, which are not required to be served by facilities - and we need to pay a penalty p_j for an unconnected city j . Algorithms for facility location with penalties have been investigated by Charikar *et al.* [14], Jain *et al.* [36]. As observed by Jain *et al.* [36], the JMS algorithm can be adapted to facility location with penalties with the following modification: If α_j reaches p_j before j is connected to any open facility, then the city j stops raising its budget and keeps its budget equal to its penalty until it is either connected to an open facility or all remaining cities stop raising their budgets. At this point, the algorithm terminates and unconnected cities remain unconnected.

Consider a star of the optimal solution, with facility f and cities $1, 2, \dots, k$ of connection costs d_1, d_2, \dots, d_k . Thus, $d_j \leq p_j$ for $j =$

$1, 2, \dots, k$. Due to the modified algorithm, we also have $\alpha_j \leq p_j$ for $j = 1, 2, \dots, k$. Thus, we obtain the following factor-revealing LP:

$$\begin{aligned}
 & \text{maximize } \sum_{j=1}^k \alpha_j & (5.25) \\
 & \text{subject to } f + \sum_{i=1}^k d_i = 1 \\
 & \alpha_i \leq \alpha_{i+1}, \quad \forall 1 \leq i < k \\
 & r_{j,i} \geq r_{j,i+1}, \quad \forall 1 \leq j < i < k \\
 & \alpha_j \geq r_{j,j+1}, \quad \forall 1 \leq j < k \\
 & \alpha_i \leq r_{j,i} + d_i + d_j, \quad \forall 1 \leq j < i \leq k \\
 & \sum_{j=1}^{i-1} [r_{j,i} - d_j]_+ + \sum_{j=i}^k [\alpha_i - d_j]_+ \leq f, \quad \forall 1 \leq i \leq k \\
 & \alpha_j \leq p_j, \quad \forall 1 \leq j \leq k \\
 & d_j \leq p_j, \quad \forall 1 \leq j \leq k \\
 & \alpha_j, d_j, f, r_{j,i} \geq 0, \quad \forall 1 \leq j \leq i \leq k.
 \end{aligned}$$

The reader can verify that this modified JMS algorithm also yields the same approximation ratio as the standard JMS (for the metric facility location problem).

5.5.4 Fault-tolerant facility location

In the fault-tolerant facility location problem, we are given a connectivity requirement q_j for each city j , specifying the number of open facilities to which city j should be connected. In some scenarios, facilities may have a failure chance, thus assigning q_j facilities to city j provides a more secure connection to each city. For example (*cf.* [72]), in a distributed network, we want to place caches and assign data requests to caches so as to be resistant against caches becoming unavailable due to node or link failures. A common solution is to repli-

cate data items across caches and build some resilience in the network. This motivates the *fault-tolerant facility location*.

We consider fault-tolerant facility location (FTFL) with uniform connectivity requirements, i.e., $q_j = q$ for all $j \in \mathcal{C}$. The algorithm can be modified as follows: We raise α_j until city j is connected to q facilities. It has been shown by Swamy and Shmoys [72] that this modified JMS algorithm also achieves the same approximation ratio as the standard JMS (for the metric facility location problem).

5.6 REMARKS AND OPEN PROBLEMS

We have seen an analysis of the worst-case performance ratio of the JMS algorithm for metric facility location problems by formulating a factor-revealing linear program (with a parameter k). The upper bound of the approximation ratio is attained for $k \rightarrow \infty$. However, it is generally hard to compute this value. Analyzing the upper bound for arbitrarily large k in a way inspired by Fernandes *et al.* [29] results in a bound which is quite close to the true approximation ratio. As we have shown in Section 5.3, given a factor-revealing LP, we aim at finding a new LP (“the upper bound factor-revealing LP”), whose optimal solution value provides an upper bound of the factor-revealing LP. Solving this upper bound factor-revealing LP for a large parameter \hat{k} , we can get a tight upper bound for the true approximation ratio. Indeed, it has been shown by Fernandes *et al.* [29] that the limits of the factor-revealing LP (5.6) and its upper bound factor-revealing LP (5.7) coincide.

A natural question would be closing the approximability gap. A possible way could be to exhibit other refinements of the JMS algorithm. Also, it is interesting to use this factor-revealing LP technique to analyze algorithms for other combinatorial optimization problems.

Summary

Combinatorial optimization problems can often be formulated as integer linear programs, which are hard to solve in general. Approximating the optimal solutions of these problems by means of LP rounding is a basic technique used in many places. As we have seen in Chapter 1, for easy problems like Maximum Weight Matching in bipartite graphs, its LP relaxation even provide optimal (integral) solutions. However, this property does not hold for all combinatorial optimization problems, *i.e.*, there is an integrality gap between the optimal fractional solution and the optimal integral solution. Studying the integrality gap (*cf.* Section 1.5) can help us to estimate the performance of LP rounding. In addition, it turns out that a small relative integrality gap yields an approximately fair allocation of profits/costs among all players in cooperative games (*cf.* Chapter 2).

6.1 RELATIVE INTEGRALITY GAPS

In Chapter 3 and Chapter 4, we analyzed the relative integrality gaps of the integer linear programs that underlie the uniform and non-uniform bin packing games respectively. The relative integrality gap is defined by the optimum integral value divided by the optimum fractional value. An instance N of the (uniform) bin packing game contains n items of sizes a_1, \dots, a_n and k bins of capacity 1 each. The value function $v(S)$ for S containing items and bins is the maximum total size of items of S that can be packed into the bins of S . Note that

the total size of packed items cannot exceed the bin capacity. An instance of a non-uniform bin packing game is similarly defined except that bins are allowed to have different capacities.

Our main contribution is an improved analysis on the relative integrality gap for bin packing games. In the uniform case, we proved that $v(N)/v'(N) \geq 3/4$ for any uniform bin packing instance N , where $v(N), v'(N)$ are optimal integral and fractional packing value respectively. This in turn implies that the $1/4$ -core is always nonempty, which improves the result due to Woeginger [74] (saying that the $1/3$ -core is always nonempty).

However, the derivation of our result is a long way. At the very first beginning, we proposed a new rounding heuristic (called greedy selection, *cf.* Section 3.2) which extracts feasible sets from an optimum fractional packing in a greedy way. Unfortunately, this approach only yields an alternative proof of non-emptiness of the $1/3$ -core. But, inspired from this, we came up with a modified greedy selection (*cf.* Section 3.3) that extracts feasible sets in a more tactical way, thereby improving the $1/3$ bound a little bit down to $1/3 - 1/108$.

We did not expect that the modified greedy selection can yield a better bound (*i.e.*, better than $1/3 - 1/108$), because it is difficult to analyze the instance with many small items (for some technical reasons). To overcome this difficulty, we presented another approach (*cf.* Section 3.5), which yields integral packings by means of finding a maximum weight matching for large items then greedily packing small items afterwards. Still, it turns out that this approach alone does not yield our result. Finally, we succeeded in combining ideas from this "matching plus greedy packing" approach with the greedy selection we had at the beginning (however with slight variants), thereby proving the claim that the $1/4$ -core is always nonempty.

In the non-uniform case, we proved that $v(N)/v'(N) \geq 1/2$ for all N , improving the result of Faigle and Kern [28] (saying that $v(N)/v'(N) \geq 1/2$ if any item fits into any bin). Since bins may have different sizes, here a feasible set must relate the individual bins. Thus, analyzing

the corresponding relative integrality gap becomes very difficult and many ideas from the uniform case even does not apply here.

In particular, we considered two special cases: 1. All item sizes are larger than $1/3$; 2. The number of bins tends to infinity. For the first case, we proved that $v(N)/v'(N) \geq 7/12$, *i.e.*, the $5/12$ -core is always nonempty, by a reversed greedy packing idea (*cf.* Section 4.3). For the second case, we extended the result in the uniform case, saying that the ϵ -core is nonempty if the number of bins is sufficiently large, to the non-uniform case. However, what slightly different is, the average bin size is taken into account in the non-uniform case.

6.2 FACILITY LOCATION

In Chapter 5, we investigated approximation algorithms for the metric facility location problem, a basic NP-hard problem in combinatorial optimization. We introduced the JMS algorithm proposed by Jain *et al.* (*cf.* Section 5.2) and analyzed a variant of the JMS algorithm, due to Mahdian *et al.* *cf.* Section 5.2.3. The approximation ratios of both algorithms can be bounded by a so-called factor-revealing LP.

Analyzing upper bounds of certain linear programs is often hard in general. The proof of approximation ratio of the JMS algorithm due to Jain *et al.* [37] involves observations from numerical experiments and quite extensive calculations. Our main contribution is an improved analysis for the derivation on upper bound of the factor-revealing LP. Our approach is inspired by Fernandes *et al.* [29], and is more direct and intuitive. Besides that, we detected and patched a small gap in earlier upper bound computations (*cf.* Lemma 5.3).

Bibliography

- [1] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristic for k-median and facility location problems. In *The thirty-third annual ACM symposium on Theory of computing*, STOC '01, pages 21–29, New York, NY, USA, 2001. ACM.
- [2] C. Berge. Balanced matrices. *Mathematical Programming*, 2:19–31, 1972.
- [3] V. Bilò. On the packing of selfish items. In *The 20th international conference on Parallel and distributed processing*, IPDPS '06, pages 45–45. IEEE Computer Society, 2006.
- [4] J. Borwein and A. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. CMS Books in Mathematics. Springer, 2005.
- [5] N. Buchbinder and J. S. Naor. The design of competitive online algorithms via a primal-dual approach. *Foundations and Trends in Theoretical Computer Science*, 3(2-3):93–263, 2007.
- [6] J. Byrka. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. In *Approximation Randomization, and Combinatorial Optimization. Algorithms and Techniques*, APPROX/RANDOM '07, Lecture Notes in Computer Science, pages 29–43, 2007.
- [7] J. Byrka and K. Aardal. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. *SIAM Journal on Computing*, 39(6):2212–2231, 2010.

- [8] J. Byrka, M. Ghodsi, and A. Srinivasan. Lp-rounding algorithms for facility-location problems. *CoRR*, abs/1007.3611, 2010.
- [9] P. Camion. Characterization of totally unimodular matrices. *Proceedings of the American Mathematical Society*, 16(5):1068–1073, 1965.
- [10] A. Caprara, H. Kellerer, and U. Pferschy. The multiple subset sum problem. *SIAM Journal of Optimization*, 11:308–319, 1998.
- [11] A. Caprara, H. Kellerer, and U. Pferschy. A $3/4$ -approximation algorithm for multiple subset sum. *Journal of Heuristics*, 9(2):99–111, 2003.
- [12] M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *The 40th Annual Symposium on Foundations of Computer Science, FOCS '99*, pages 378–388, 1999.
- [13] M. Charikar, S. Guha, É. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *J. Comput. Syst. Sci.*, 65(1):129–149, 2002.
- [14] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *The 12th annual ACM-SIAM symposium on Discrete algorithms, SODA '01*, pages 642–651. Society for Industrial and Applied Mathematics, 2001.
- [15] M. Charikar and S. Li. A dependent lp-rounding approach for the k-median problem. In *The 39th international colloquium conference on Automata, Languages, and Programming - Volume Part I, ICALP '12*, Lecture Notes in Computer Science, pages 194–205. Springer-Verlag, 2012.
- [16] F. A. Chudak. Improved approximation algorithms for uncapacitated facility location. In *Integer Programming and Combinatorial*

- Optimization, IPCO '98*, Lecture Notes in Computer Science, pages 180–194. Springer, 1998.
- [17] F. A. Chudak and D. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM Journal on Computing*, 33(1):1–25, 2003.
- [18] M. Conforti, G. Cornuéjols, and K. Vušković. Balanced matrices. *Discrete Mathematics*, 306(19-20):2411–2437, 2006.
- [19] S. A. Cook. The complexity of theorem-proving procedures. In *The third annual ACM symposium on Theory of computing, STOC '71*, pages 151–158, 1971.
- [20] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial optimization*. John Wiley & Sons, Inc., New York, NY, USA, 1998.
- [21] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 2009.
- [22] G. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1998.
- [23] F. de la Vega and G. Lueker. Bin packing can be solved within $1 + \epsilon$ in linear time. *Combinatorica*, 1:349–356, 1981.
- [24] J. Edmonds. Paths, trees, and flowers. *Canad. J. Math.*, 17:449–467, 1965.
- [25] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1:127–136, 1971.
- [26] L. Epstein and E. Kleiman. Selfish bin packing. *Algorithmica*, 60:368–394, 2011.
- [27] U. Faigle and W. Kern. On some approximately balanced combinatorial cooperative games. *Methods and Models of Operation Research*, 38:141–152, 1993.

- [28] U. Faigle and W. Kern. Approximate core allocation for binpacking games. *SIAM J. Discrete Math*, 11:387–399, 1998.
- [29] C. Fernandes, L. Meira, F. Miyazawa, and L. Pedrosa. A systematic approach to bound factor revealing LPs and its application to the metric and squared metric facility location problems. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM '12*, volume 7408 of *Lecture Notes in Computer Science*, pages 146–157. Springer, 2012.
- [30] D. Fulkerson, A. Hoffman, and R. Oppenheim. On balanced matrices. In *Pivoting and Extension*, volume 1 of *Mathematical Programming Studies*, pages 120–132. Springer Berlin Heidelberg, 1974.
- [31] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.
- [32] M. X. Goemans and M. Skutella. Cooperative facility location games. *Journal of Algorithms*, pages 194–214, 2004.
- [33] D. Granot and G. Huberman. Minimum cost spanning tree games. *Mathematical Programming*, 21:1–18, 1981.
- [34] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of Algorithms*, 31(1):228 – 248, 1999.
- [35] D. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22:148–162, 1982.
- [36] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *Journal of the ACM*, 50(6):795–824, 2003.
- [37] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *The 34th annual ACM symposium on Theory of computing, STOC '02*, pages 731–740, 2002.

- [38] K. Jain and V. V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *The 40th Annual Symposium on Foundations of Computer Science, FOCS '99*, pages 2–13, Washington, DC, USA, 1999. IEEE Computer Society.
- [39] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.
- [40] B. Joosten. Relaxation of 3-partition instances. Master's thesis, University of Twente, 2011.
- [41] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–396, 1984.
- [42] R. M. Karp. Reducibility among combinatorial problems. In *50 Years of Integer Programming 1958-2008*, pages 219–241. Springer Berlin Heidelberg, 2010.
- [43] W. Kern and X. Qiu. Improved taxation rate for bin packing games. In *First International ICST Conference on Theory and Practice of Algorithms in (Computer) Systems, TAPAS '11*, volume 6595 of *Lecture Notes in Computer Science*, pages 175–180. Springer Verlag, 2011.
- [44] W. Kern and X. Qiu. Improved taxation rate for bin packing games (extended abstract). In *The 10th Cologne-Twente Workshop on graphs and combinatorial optimization, CTW 2011*, pages 173–176, 2011.
- [45] W. Kern and X. Qiu. Integrality gap analysis for bin packing games. *Operations Research Letters*, 40(5):360 – 363, 2012.
- [46] W. Kern and X. Qiu. Note on non-uniform bin packing games. *Discrete Applied Mathematics*, 2012. In press.
- [47] W. Kern and X. Qiu. The $1/4$ -core of the uniform bin packing game is nonempty. In *The 19th Annual International Computing*

- and Combinatorics Conference, COCOON '13*, volume 7936 of *Lecture Notes in Computer Science*, pages 41–52. Springer Berlin Heidelberg, 2013.
- [48] V. Klee and G. J. Minty. How good is the simplex algorithm? In *Inequalities*, volume III, pages 159–175. Academic Press, New York, 1972.
- [49] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.
- [50] J. Kuipers. Bin packing games. *Mathematical Methods of Operations Research*, 47:499–510, 1998.
- [51] R. E. Ladner. On the structure of polynomial time reducibility. *J. ACM*, 22(1):155–171, 1975.
- [52] S. Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. In *The 38th international conference on Automata, languages and programming - Volume Part II, ICALP'11*, *Lecture Notes in Computer Science*, pages 77–88. Springer, 2011.
- [53] S. Li and O. Svensson. Approximating k-median via pseudo-approximation. *CoRR*, abs/1211.0243, 2012.
- [54] Z. Liu. Complexity of core allocation for the bin packing game. *Operations Research Letters*, 37(4):225 – 229, 2009.
- [55] L. Lovász and M. D. Plummer. *Matching Theory*. North-Holland Math, Amsterdam, 1986.
- [56] M. Mahdian and Q. Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing LPs. In *The 43rd annual ACM symposium on Theory of computing, STOC '11*, pages 597–606, 2011.

- [57] M. Mahdian, Y. Ye, and J. Zhang. Improved approximation algorithms for metric facility location problems. In *The 5th International Workshop on Approximation Algorithms for Combinatorial Optimization, APPROX '02*, Lecture Notes in Computer Science, pages 229–242, 2002.
- [58] T. Matsui. A minimum taxrate core allocation of bin packing games. In *Game Theory and Applications*, volume V, pages 73–79. Nova Science Pub Inc, 2000.
- [59] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: Algorithms and complexity*. Dover publications, Inc., Mineola, New York, 1998.
- [60] C. M. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
- [61] B. Peleg and P. Sudhölter. *Introduction to the Theory of Cooperative Games (Theory and Decision Library C)*. Springer, 3rd edition, 2007.
- [62] R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technology Journal*, 36:1389–1401, 1957.
- [63] G. Schäfer. Lecture notes of the master course: Discrete optimization, 2012. Utrecht University.
- [64] D. Schmeidler. The nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics*, 17(6):1163–1170, 1969.
- [65] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, Inc., New York, NY, USA, 1998.
- [66] L. Shapley and M. Shubik. The assignment game i: The core. *International Journal of Game Theory*, 1:111–130, 1971.
- [67] L. S. Shapley. A value for n-person games. *Contributions to the theory of games*, 2:307–317, 1953.

- [68] L. S. Shapley. On balanced sets and cores. *Naval Research Logistics Quarterly*, 14:453–460, 1967.
- [69] D. B. Shmoys, E. Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *The 29th annual ACM symposium on Theory of computing*, STOC '97, pages 265–274, 1997.
- [70] M. Sipser. *Introduction to the Theory of Computation*. International Thomson Publishing, 1st edition, 1996.
- [71] M. Sviridenko. An improved approximation algorithm for the metric uncapacitated facility location problem. In *Integer Programming and Combinatorial Optimization, IPCO '06*, volume 2337 of *Lecture Notes in Computer Science*, pages 240–257, 2006.
- [72] C. Swamy and D. B. Shmoys. Fault-tolerant facility location. In *The 14th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '03*, pages 735–736, 2003.
- [73] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, USA, 2011.
- [74] G. J. Woeginger. On the rate of taxation in a cooperative bin packing game. *Mathematical Methods of Operations Research*, 42:313–324, 1995.
- [75] G. Yu and G. Zhang. Bin packing of selfish items. In *Internet and Network Economics*, volume 5385 of *Lecture Notes in Computer Science*, pages 446–453. Springer Berlin Heidelberg, 2008.

Publications

- [KQ11a] Walter Kern and Xian Qiu. Improved taxation rate for bin packing games. In *First International ICST Conference on Theory and Practice of Algorithms in (Computer) Systems, TAPAS 2011*, volume 6595 of *Lecture Notes in Computer Science*, pages 175–180. Springer Verlag, 2011.
- [KQ11b] Walter Kern and Xian Qiu. Improved taxation rate for bin packing games (extended abstract). In *The 10th Cologne-Twente Workshop on graphs and combinatorial optimization, CTW 2011*, pages 173–176, 2011.
- [KQ12a] Walter Kern and Xian Qiu. Integrality gap analysis for bin packing games. *Operations Research Letters*, 40(5):360 – 363, 2012.
- [KQ12b] Walter Kern and Xian Qiu. Note on non-uniform bin packing games. *Discrete Applied Mathematics*, 2012. In press.
- [KQ13a] Walter Kern and Xian Qiu. The $1/4$ -core of the uniform bin packing game is nonempty. In *The 19th Annual International Computing and Combinatorics Conference, COCOON 2013*, volume 7936 of *Lecture Notes in Computer Science*, pages 41–52. Springer Berlin Heidelberg, 2013.
- [KQ13b] Walter Kern and Xian Qiu. An improved analysis to the JMS algorithm with scaling and augmentation. 2013. Working paper.