

A Unified Race Algorithm for Offline Parameter Tuning

Tim van Dijk, Martijn Mes, Marco Schutten, Joaquim Gromicho

Beta Working Paper series 453

BETA publicatie	WP 453 (working paper)
ISBN	
ISSN	
NUR	982
Eindhoven	April 2014

A Unified Race Algorithm for Offline Parameter Tuning

Tim van Dijk^a, Martijn Mes^{a,*}, Marco Schutten^a, Joaquim Gromicho^b

^a*Department Industrial Engineering and Business Information Systems, School of Management and Governance, University of Twente, The Netherlands*

^b*Department Algorithmic Innovation, ORTEC Software Development, Gouda, The Netherlands*

Abstract

This paper proposes uRace, a unified race algorithm for efficient offline parameter tuning of deterministic algorithms. We build on the similarity between a stochastic simulation environment and offline tuning of deterministic algorithms, where the stochastic element in the latter is the unknown problem instance given to the algorithm. Inspired by techniques from the simulation optimization literature, uRace enforces fair comparisons among parameter configurations by evaluating their performance on the same training instances. It relies on rapid statistical elimination of inferior parameter configurations and an increasingly localized search of the parameter space to quickly identify good parameter settings. We empirically evaluate uRace by applying it to a parameterized algorithmic framework for loading problems at ORTEC, a global provider of software solutions for complex decision-making problems, and obtain competitive results on a set of practical problem instances from one of the world's largest multinationals in consumer packaged goods.

1. INTRODUCTION

In this paper, we focus on efficient offline parameter tuning of deterministic algorithms with parameter spaces that may be enormous and include categorical parameters. The performance of a parameterized algorithm on specific problem instances depends on the quality of its parameter configuration. Moreover, the same parameter settings likely perform differently on distinct problem instances. This suggests the need for a tuning algorithm that quickly identifies good parameter settings for a parameterized algorithm such that its performance on a specific type of instance is optimized. We propose an innovative unified race algorithm (uRace) to satisfy this need.

The development of uRace finds its origin at ORTEC, a global provider of planning and optimization software solutions. One of their software solutions is ORTEC Pallet and Load Building (OPLB), which aims to facilitate efficient pallet and container loading for customers worldwide. The list of customers for OPLB includes many well-known multinationals, each of which use the software to find good loading plans that optimize their specific objective criterion while satisfying all constraints. Figure 1 shows examples of the resulting load visualizations for two different applications.

*Corresponding author. E-mail: m.r.k.mes@utwente.nl; Tel.: +31 534894062; Fax: +31 534892159.

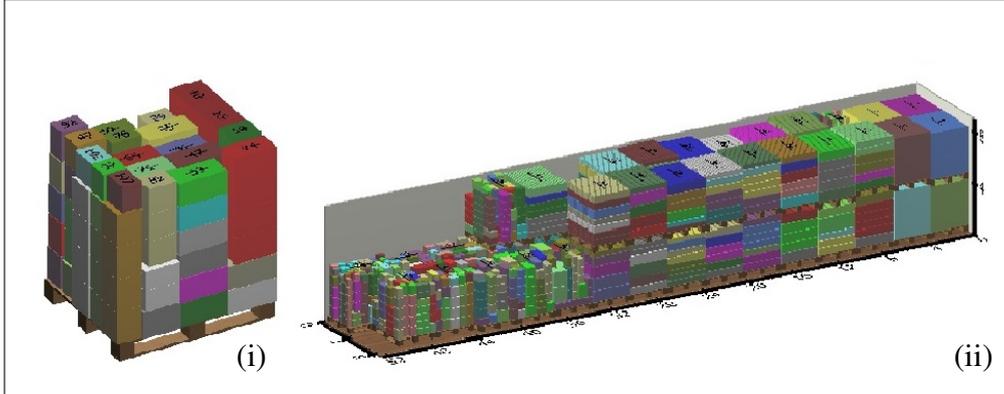


Figure 1: Example of a visualization of (i) a load of boxes on a pallet and (ii) a load of (pick)pallets in a container.

OPLB contains a heavily parameterized algorithmic framework that consists of two parts: (i) a greedy construction phase, which incrementally and deterministically constructs a solution, and (ii) a local search phase, which iteratively destroys several of the last executed item placements, selects an alternative item placement at random, and subsequently reconstructs using the same greedy construction. Particularly the greedy construction phase can be customized with much degree of freedom by tuning its parameters. The parameters are mainly categorical sorting parameters that sort the options for item placement on various preferences, e.g., sorting the remaining items on “length” in “ascending” order, or the empty subspaces in the container on “volume” in “descending” order. In total, there are approximately 6×10^{39} possible parameter configurations. The item placement preference lists, and therefore the parameter settings in the greedy construction phase, govern the incremental construction. As a result, these parameter settings determine the quality of the initial solution. Moreover, as a result of the neighborhood structure in the local search algorithm, the quality of the initial solution translates well to the quality of OPLB’s final solution. For this reason, we exclude the local search from our tuning effort, as this radically speeds up our measurements and removes any stochastic noise from our observations, which greatly benefits the learning process of our tuning algorithm.

We distinguish two different approaches through which we may seek optimal parameter settings for the greedy construction phase of OPLB. First, we can opt for problem instance tuning, in which case we tune the parameters for the specific problem instance the customer currently faces, the online problem. Second, we can opt for problem family tuning, also denoted by offline tuning, in which case we tune the parameters for a problem the customer *typically* faces, simulated with a set of representative training instances. In essence, problem instance tuning is an extreme of problem family tuning where the training set coincides with the online problem, hence produces better results in case the required computation time is not an issue. However, for problem instance tuning in our practical setting, the computationally expensive tuning process occurs every time a customer wants to load a container, which imposes severe limitations on the tuning time. Indeed, any self-learning tuning process would be computationally problematic for problem instance tuning, and our enormous search space makes complete enumeration computationally prohibitive. Therefore, we prefer to tune the parameters before the customer wants to load a container, whenever computational resources are amply available, on a set of training instances (e.g., recent historic instances from that customer). The resulting parameter configura-

tion can subsequently be applied to any occurring problem instance at the customer site without any further tuning.

Evidently, problem family tuning is only viable when a configuration that performs well on the training set also performs well on similar, yet unseen problems. To verify this, we uniformly draw a sample of parameter configurations from our parameter space and evaluate their performance on a set of similar loading problems from one of ORTEC’s customers. Figure 2 illustrates two effects that suggest that such a generalization of quality holds for parameter configurations in OPLB. The graph on the left shows the relation between the average performance on a training set from the customer and the performance on one, randomly selected, other instance of this customer. The strong positive correlation implies that, provided we have a representative training set, the average performance on the training set is a good indication of the performance on the online problem. Moreover, the graph on the right shows the relation between the average performance on the complete set of instances and the coefficient of variation (CV) of the performance across the set, where a low CV indicates a high robustness. When looking at the correlation between the average and variance in volume utility, we observe a strong correlation (Pearson coefficient of -0.87 , p -value = 0.000), which indicates that while optimizing the average performance of a parameter configuration on the training set, we simultaneously optimize its robustness, and therefore the extent to which its quality generalizes to the online problem. These are crucial results for our research, as they advocate the use of problem family tuning.

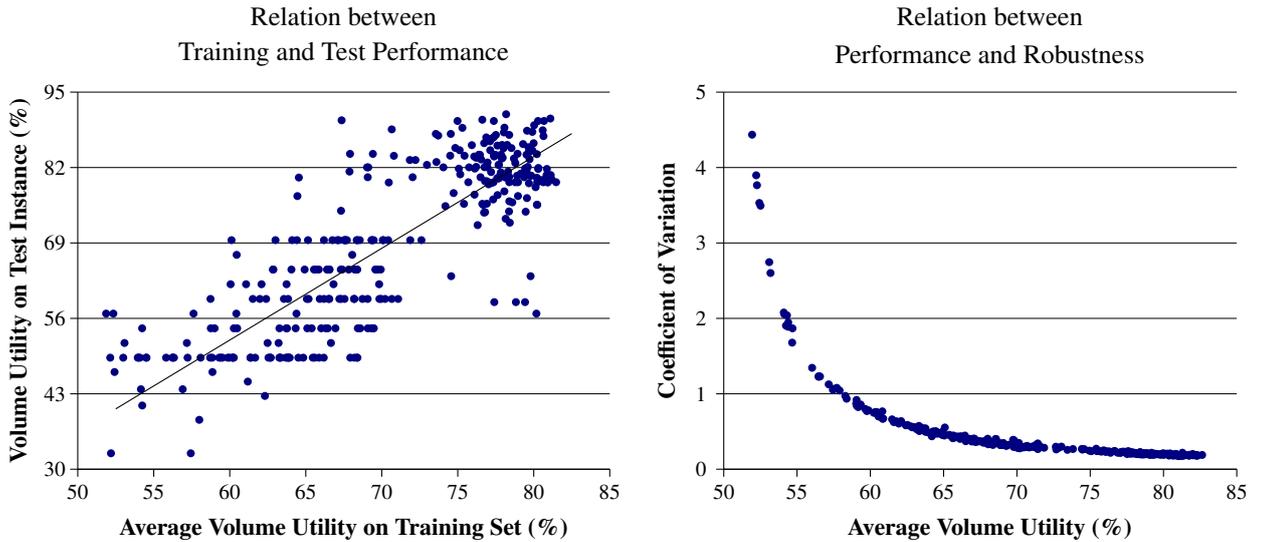


Figure 2: Illustration of the viability of problem family tuning of the parameters in OPLB. Each dot represents one randomly sampled parameter configuration. The performance is expressed as the volume utility (%) of the loaded container. The coefficient of variation is expressed as the ratio of the sample standard deviation to the sample mean of the performance distribution on the training set.

The contribution of this paper is the development of uRace, a unified race algorithm for efficient offline parameter tuning of deterministic algorithms. We apply uRace to a parameterized loading algorithm within OPLB. On a set of practical problem instances from a multinational in consumer packaged goods, we obtain competitive results compared to well known algorithms

in the literature. Moreover, as a result of the similar stochastic element involved in stochastic simulations as in problem family tuning, uRace can also be used for simulation optimization purposes.

The remainder of this paper is structured as follows. In Section 2, we present some notable work that aids in the development of a problem family tuning method. In Section 3, we present uRace. In Section 4, we provide the results of our computational experiments. Finally, in Section 5, we provide our conclusions regarding theoretical and practical relevance, and offer suggestions for further research.

2. PROBLEM SETTING

In the field of parameter tuning, we are interested in “a staggeringly difficult yet beautifully concise problem” (Lizotte, 2008):

$$\arg \max_{\theta \in \Theta} f(\theta) \tag{1}$$

When solving Equation 1, the aim is to quickly identify, from a set of alternatives Θ , the one alternative θ^* that gives the highest (expected) return. This problem appears in many forms in the literature. In the field of simulation optimization, the function $f(\theta) = \mathbb{E}[F(\theta, \omega)]$ follows from a simulation run with sample path ω and sample performance $F(\theta, \omega)$ (Fu et al., 2005). For example, $F(\theta, \omega)$ gives the simulated profit of a shop, considering stochastic customer choice and demand, where the corresponding set Θ would be the possible choices in product variants to stock (Hong and Nelson, 2009).

The literature covers many relevant techniques to solve Equation 1. In sequential measurement techniques, for example, the search for the best alternative is sequentially guided towards the next measurement depending on previous observations. An example of a relatively simple sequential measurement technique is an upper confidence bound (UCB) policy, in which, consistently, the alternative with the highest UCB in terms of return is measured, resulting in continuously shrinking performance distributions (Chang et al., 2007). Typically, such simple heuristics require each alternative to be measured at least once, which becomes computationally prohibitive for larger parameter spaces.

More sophisticated techniques are able to tackle larger search spaces. In such approaches, a priori knowledge on interrelations in the parameter space is used to design correlated beliefs about the effect of the measured performance of some alternative on the expected performance of some other alternative, such that estimates can be given even for alternatives that have not been measured before. Such extrapolation of the observations on one alternative to the beliefs of other alternatives is called generalization. Most sequential measurement policies that are applicable to large search spaces rely heavily on generalization techniques. For example, the knowledge gradient policy (Frazier et al., 2008) myopically maximizes the expected increment in the value of information by continuously directing the walk towards the point that will reveal the most information on other unseen points. While the updated knowledge gradient policy by the same authors assumes a priori knowledge on correlations in the belief structure of Θ , the hierarchical knowledge gradient (Mes et al., 2011) requires a known aggregation structure of the parameter space, which is generally easier to obtain. In the related field of Bayesian global optimization of continuous functions, one would place a prior belief on the function f and relies on the laws of probability to update these beliefs to posterior distributions based on accumulated observed data. The response-surface methodology (RSM) (Law and Kelton, 2000), e.g., takes a Bayesian

approach to fit a second order response surface to a series of experiments, conducted using a simple experimental design, and subsequently finds the optimal point on this surface. Most sequential measurement policies are designed for stochastic experiments with noisy measurements, but some can be applied to deterministic cases, and others are designed specifically for noise-free measurements. The efficient global optimization (EGO) method (Jones et al., 1998), e.g., assumes deterministic measurements and attempts to fit a smooth, differentiable function through previous measurements while continuously selecting points based on expected improvement of the objective function.

Our specific tuning problem possesses a few important characteristics that determine the suitability of tuning techniques. First, our parameter space, hence our search space of alternatives Θ , is enormous, which eliminates the use of any tuning technique that requires a full factorial design or needs to rank all possible decisions before making the decision. Second, our parameter space is of categorical nature, such that $\theta \in \Theta$ is a vector of categorical attributes, which complicates the use of generalization techniques. Techniques such as RSM and EGO assume a spatial relationship between the parameter space on the one hand, and a supposedly smooth performance landscape on the other hand, which they attempt to exploit to guide the tuning process. In many practical applications with numerical variables, this assumption is reasonable. However, for categorical variables, which only have a symbolic meaning and do not have a natural way to be encoded, the assumption of a regular or smooth performance landscape is not likely to hold. Indeed, it is not trivial to quantify the distance between categorical parameter values (e.g., sorting items by volume or by priority), let alone to assume that this unquantifiable distance corresponds with similar distances in the performance landscape. Third, recall that our problem family tuning process occurs on a set of training instances, which we denote by \mathcal{X} , prior to knowing the online problem instance at the customer site. That is, we effectively simulate reality (the unseen online problem instance) with similar, representative situations (a training set with other problem instances in the same problem family), hence introducing a stochastic element to our tuning process. We obtain information about a configuration θ by evaluating its quality on a training instance x_i in \mathcal{X} . Finally, $f : \Theta \rightarrow \mathbb{R}$ is an algorithm that translates the input vector into a single value, indicating the solution quality of the resulting loading plan on a particular problem instance, where every individual measurement $f(\theta|x_i)$ is deterministic.

Our objective is to identify the parameter configuration θ that maximizes the expected value of f on \mathcal{X} , which, as Figure 2 suggests, coincides with our ultimate objective to optimize the loading solution on the online problem instance at the customer site. We obtain the offline algorithm configuration problem (Birattari, 2009):

$$\arg \max_{\theta \in \Theta} \mathbb{E}_{\mathcal{X}} [f(\theta)] = \frac{1}{|\mathcal{X}|} \arg \max_{\theta \in \Theta} \sum_{i=1}^{|\mathcal{X}|} f(\theta|x_i). \quad (2)$$

Tuning methods specifically designed to efficiently solve Equation 2 are race algorithms. In a race algorithm, parameter configurations are evaluated in batches, on a sequence of problem instances, with the aim to eliminate candidate configurations based on statistical inferiority, such that an increasing number of configurations “drop out of the race”. This speeds up the tuning process and allows more productive allocation of computation time to the most promising configurations.

One of the first successful race algorithms, named F-Race, is presented by Birattari et al. (2002). F-Race starts with a set Θ_r of candidate configurations from which it tries to exclude inferior members as soon as sufficient statistical evidence is collected until either one configu-

ration remains or some predefined computation budget is exhausted. An obvious drawback of F-Race is that the winning configuration is always a member of the initial set Θ_r . Therefore, Θ_r need be obtained through a full factorial design, or at least be a large subset of Θ , which becomes computationally prohibitive for large parameter spaces. To tackle this issue, Balaprakash et al. (2007) propose an extension of F-Race, iterative F-Race (I/F-Race), which consists of multiple, iterated applications of F-Race. In the first iteration, a multi-variate uniform distribution is used to randomly sample the initial set of candidate configurations Θ_r . In each subsequent iteration, a new set Θ_r is constructed based on the most promising configurations from the previous iteration. Through the iterated nature of a race algorithm, the requirement of a full factorial design is removed. Rather than requiring information about all members of Θ , a subset $\Theta_r \in \Theta$ is used that gradually changes in composition. A suitable technique to reconstruct the set Θ_r based on previous observations, in light of the categorical parameters in OPLB, is the use of biased sampling distributions (Birattari et al., 2002). This technique does not require the existence of a smooth and predictable performance landscape nor a full factorial design, and randomly samples parameter settings for new measurements, while continuously adapting the sampling distributions based on previous measurements. For categorical parameters, new values are sampled from discrete sampling distributions with a bias towards the parameter settings from known configurations that have shown good performance, the remaining “racers” from the previous iteration. This bias can be increased throughout the tuning process to facilitate an increasingly localized search.

We apply our tuning algorithm to loading problems. All loading problems share an identical structure, which can be summarized as follows. We have two sets of elements, a set of large objects (e.g., pallets or containers) and a set of small items that need to be placed in or on these large objects. The small items are defined in one, two, or three geometric dimensions (and may possess other characteristics such as weight and value). The loading problem is solved by selecting small items, grouping them into one or more subsets, and assigning each of them to one of the large objects such that a given objective function is optimized and the following geometric restrictions are met: (a) all small items of the subset lie entirely within the large object, and (b) the small items do not overlap. Other constraints may apply on top of these, e.g., fixed orientations and support, axle weight, and stackability constraints. The literature covers a wide scope of solution methods to tackle loading problems. For example, George and Robinson (1980) propose a wall building approach in which the large object is filled with vertical cuboid layers, “walls”, along the width of the container. Bischoff et al. (1995) propose a horizontal layer approach, in which the container is filled with horizontal cuboid layers that each cover the largest possible fraction of the surface underneath. For a recent state-of-the-art review on loading problems, we refer to Bortfeldt and Wäscher (2013).

3. DESIGN OF A TUNING ALGORITHM

Our tuning algorithm is inspired by I/F-Race (see Section 2). However, we propose an important structural change to utilize obtained information more efficiently. As said, I/F-Race enforces fair comparisons, by letting each iteration be an autonomous F-Race. Information from previous iterations is used to generate a set of configurations to commence the next F-Race, but the exact results of individual measurements are discarded. To enable the use of this valuable information, while still ensuring fair comparisons, we prefer to first evaluate every newly generated configuration on the same problem instances as those previously solved by its parent configuration. Since this removes the stochastic element inherent to problem instance selection, this gives us more information about the new configurations than measurements on new problem instances would.

The result of this structural change is that we create one unified race algorithm, rather than a sequence of autonomous F-Race algorithms. We consider each iteration to be a batch measurement of all candidate racers on a single problem instance, and we generate new configurations immediately whenever statistically inferior configurations are eliminated in our sample. This allows for more natural allocation of computation time to either exploration (i.e., exploring new points in the solution space) or exploitation (i.e., acquiring more reliable information on known solution points). Indeed, at the start of the tuning process the configurations are likely relatively diverse in their performance, so we expect many inferior configurations, naturally allocating computation time to exploration. As the tuning process proceeds, and we approach the best configuration θ^* , we expect the candidates to be closer to one another in terms of performance. Therefore, by only exploring new options in case of statistical inferiority, computation time allocation naturally tends towards exploitation.

The basic structure of uRace is presented in Algorithm 1. Each iteration j starts with a set of “racing” configurations Θ_r , where we let the designated size C_τ of Θ_r depend on the fraction τ of computation budget spent and the size of our parameter space (see Section 3.3). In each iteration, all configurations in Θ_r are tested for statistical inferiority based on previous observations. If no statistical inferiority is detected, all configurations are measured on a new problem instance x_j , which is drawn randomly from the training set \mathcal{X} without replacement (as long as $|\mathcal{X}| > 0$). Any inferior configuration is eliminated, and immediately replaced by a newly generated child configuration θ^{child} . This newly generated configuration is first measured on the same stream of problem instances as its parent, as long as it is not found to be statistically inferior. Once we have generated sufficient non-inferior child configurations such that the set of racing configurations Θ_r is back to its designated size, we continue with a new iteration. When all problem instances have been used ($|\mathcal{X}| = 0$) and Θ_r has its designated size, we remove the worst configuration from Θ_r (even when this configuration is not statistically inferior) and continue exploration.

The tuning algorithm requires as input a set of training instances \mathcal{X} , the parameter space \mathcal{Y} , and the underlying algorithmic framework f . The user is free to set the tuning budget B , enabling easy regulation of the computation time spent on the tuning procedure. Moreover, we have two tunable parameters, α and β , that can be used to affect the exploration-exploitation tradeoff and the locality of the search. We return to these parameters in Section 3.2 and Section 3.3, respectively.

The remainder of this section describes uRace in detail, covering the three main components of an iterated race algorithm. First, Section 3.1 discusses the construction of the initial set Θ_r . Second, Section 3.2 discusses the statistical test that detects inferior members within Θ_r . Third, Section 3.3 discusses the method to construct subsequent (sets of) candidate configurations based on the remaining configurations from the previous iteration.

3.1. Initial Generation

To compose our initial set of parameter configurations, we need to select a set $\Theta_r \subset \Theta$. The initial set of configurations forms the roots of all successively generated parameter configurations, so it is desirable that it includes points from all regions in the parameter space. Latin Hypercube Sampling (LHS), first proposed by McKay et al. (1979), is a method to construct such a representative subset by drawing a sample that uniformly covers the entire multi-dimensional parameter space. LHS selects n values from k parameters by dividing the range of each parameter into n non-overlapping, equally spaced intervals, which is problematic for categorical parameters. To adapt the traditional LHS to be applied to categorical parameters, we propose a straightforward mapping structure. That is, instead of uniformly drawing a sample of size n directly from

Algorithm 1 Basic Structure of uRace

```
input
 $\mathcal{X}$  : training set
 $\mathcal{Y}$  : parameter space
 $f(\theta, x_i) : \mathcal{Y} \rightarrow \mathbb{R}$  : algorithm

tunable parameters
 $B$  : tuning budget
 $\alpha$  : level of significance
 $\beta$  : degree of bias

initialization
 $\Theta_r \sim \text{LHS}(\mathcal{Y}, C_0)$ 
 $j \leftarrow 1$ 

output
 $\theta^*$  : best configuration in  $\Theta$  on  $\mathcal{X}$ 

while tuning budget left do
     $\Theta_r \leftarrow \Theta_r \setminus \Theta_{\text{inferior}, \alpha}$ 
    while tuning budget left and  $|\Theta_r| < C_\tau$  do
         $\theta^{\text{child}} \sim \text{Regenerate}(\mathcal{Y}, \Theta_r, \beta)$ 
        for every instance  $x_i$  solved by parent do
            evaluate  $f(\theta^{\text{child}}|x_i)$ 
        end for
        if  $\theta^{\text{child}}$  non-inferior then  $\Theta_r \leftarrow \Theta_r \cup \theta^{\text{child}}$ 
        end while
    while  $|\Theta_r| > C_\tau$  do
        eliminate worst  $\theta \in \Theta_r$ 
    end while
    if  $|\mathcal{X}| > 0$  then
        for all  $\theta \in \Theta_r$  do evaluate  $f(\theta|x_j)$ 
         $j \leftarrow j + 1$ 
    else
        eliminate worst  $\theta \in \Theta_r$ 
    end if
    end while
```

the k categorical parameter spaces, we use a single, continuous, one-dimensional space $[0, n]$. From this space, we randomly draw n values, one from each integer interval. Subsequently, we map the drawn values to each of the k categorical parameter spaces, and randomly combine the resulting categorical values with the values of other parameters to obtain n k -dimensional input vectors, our initial set of configurations. Using this categorical adaptation of LHS, we enforce an even spread of our initial sample over our entire multi-dimensional parameter space.

3.2. Statistical Inferiority Test

As soon as sufficient statistical evidence is collected that a configuration is inferior to at least one other configuration in Θ_r , we eliminate the inferior configuration. As said, we enact a batch measurement policy, where we let all racers solve the same sequence of training instances. An analysis of our problem space reveals that we cannot rely on the usual assumptions of normality and equal variances. Therefore, we employ a non-parametric alternative for the pairwise Student t -test: the paired signed-rank test, proposed by Wilcoxon (1945). We perform one-tailed tests, and compare, after every batch measurement, the best configuration (highest mean performance) with all others.

The level of significance α determines the severity of the test. The higher we set α , the less evidence we require to exclude configurations. This speeds up the procedure, but also increases the probability of falsely rejecting the null hypothesis or, in other words, eliminating non-inferior configurations. An analysis of our problem space suggest that the robustness of the configurations is of such magnitude that even for relatively high levels of significance (i.e., $0.1 \leq \alpha \leq 0.3$), unlawful rejection of the null hypothesis hardly occurs.

Moreover, as a result of the structure of uRace, the value of α also has an important effect on the aforementioned exploration-exploitation tradeoff. As said, the structure of uRace enforces that the tradeoff naturally tends towards exploitation as the tuning process proceeds, but the value of α determines the severity of this transition. For example, in the limit that $\alpha \rightarrow 0$, we can only obtain sufficient statistical evidence by determining the configuration’s true mean performance. That means that configurations are never eliminated, new configurations are never evolved, our algorithm converges to exhaustive evaluation, and all computation time is undesirably spent on exploitation of the initial sample. On the other hand, as $\alpha \rightarrow 1$, our one-tailed tests ensure that all configurations (except the very best one) are continuously eliminated. That means that all computation time is spent on exploration, and we undesirably end up making an implementation decision based on measurement on a single problem instance. Initial tests show that a good exploration-exploitation tradeoff for our parameter space can be achieved using $\alpha = 0.20$.

3.3. Regeneration

Every time a statistically inferior configuration is eliminated, a newly generated child configuration, which resembles some known, well-performing parent configuration, takes its place. This regeneration procedure consists of two parts.

First, we sample, from the set of remaining configurations Θ_r , one parent configuration θ^{parent} . We start with a uniform distribution over all remaining configurations (following our explorative search), and assign increasing probability to the very best configurations (increasingly biasing our search towards the most promising areas in our parameter space) as the tuning process proceeds.

Second, from the parent, we evolve one child configuration θ^{child} , for which we use the biased sampling distribution method (see Section 2). Judging from the high degree of synergy in our parameter space, we should not use a single distribution to sample all child configurations. Such a method is blind for the parameter interactions that are abundant in our parameter space. Instead, we propose to assign individual sampling distributions to every parent configuration, which we bias based on the settings of that configuration and its ancestors only, such that every configuration carries a sampling distribution over \mathcal{Y} , and we sample a child configuration using its selected parent’s distribution. We start with a k -variate uniform distribution over each parameter space, and for each configuration we evolve a discrete k -variate sampling distribution biased towards the settings taken by itself and its ancestors. The strength of the bias determines how locally we search around known configurations. We bias the sampling distributions such that a child configuration is most likely to inherit half of the parameters from its parent at the start of the tuning process (a relatively explorative search), and all but one of the parameters towards the end of the tuning process (i.e., increasingly localizing the search around the selected parent configuration). In between, the bias increases with τ^β , where τ is the fraction of tuning time spent and $\beta > 0$ is the degree of bias we impose. Note that, as the tuning process proceeds, τ^β increases from 0 to 1. The value for β determines the shape of this gradual process. Linear, exponential, and logarithmic functions are acquired by letting $\beta = 1$, $\beta > 1$, and $0 < \beta < 1$ respectively. Initial tests show that a good locality of search in our parameter space can be achieved with an S-shaped function acquired by using the function $\beta(\tau) = 4(1 - \tau)^2$.

We let the designated size C_τ of Θ_r depend on the size of our parameter space and the remaining tuning time. We set $C_0 = k$ at the start of uRace, where k is the number of dimensions in our parameter space \mathcal{Y} , and let it gradually decrease to 2 as the algorithm proceeds, using $C_\tau = k - \tau^{\beta(\tau)}(k - 2)$. By letting the size of Θ_r decrease, we not only increasingly localize our search around the select parent configuration, but also increasingly bias our search towards

the most promising parent configurations, as newly generated child configurations evolve from a smaller number of the best remaining configurations. As mentioned before, if at some point during the tuning process, the current size of Θ_t exceeds its designated size while no inferior racers are detected, the worst configurations are eliminated.

To summarize this section, Table 1 provides an overview of the settings we use for uRace.

Table 1: The components of uRace, the unified race algorithm we propose to develop.

Component	Method
Initial Sampling	Latin Hypercube Sampling
Number of Parallel Configurations	$C_\tau = k - \tau^{\beta(\tau)} (k - 2)$
Measurement Policy	Batch Measurements
Statistical Inferiority Test	Wilcoxon Paired Signed-Rank Test with $\alpha = 0.2$
Parent Selection	Rank-Based Sampling with Increasing Bias
Child Generation	Biased Sampling Distributions for Individual Configurations with Increasing Bias
Degree of Bias	$\beta(\tau) = 4(1 - \tau)^2$

4. RESULTS

In this section, we provide the results from our empirical evaluation of the performance of uRace that we obtained by letting it tune the parameters of OPLB and subsequently solving a set of loading problems.

The loading problems we evaluate with OPLB are practical problem instances from one of the world’s largest multinationals in consumer packaged goods. They are input minimization problems, which means that the full set of small items needs to be accommodated whilst minimizing the selection of large objects utilized. The small items are first loaded onto pick-pallets (i.e., pallets with multiple individual items in any composition) or stock-pallets (i.e., pallets with a fixed load), and the pallets are subsequently loaded into containers. The primary objective is to minimize the number of containers that are required to load all items. Secondary objectives are lower pallet usage, higher stability (i.e., the ratio of the top surface versus the bottom surface of the pick-pallets), and shorter running time.

We compare OPLB with its parameters tuned with uRace with the two well-known construction algorithms from the loading literature that are described in Section 2. Essentially, these algorithms can be seen as specific instances of OPLB, where the parameters are tuned manually by the authors of the respective papers. Hence, the main difference with our parameter configuration is that we tuned the parameters automatically with uRace. We tuned the parameters on a training set of similar, yet distinct loading problems from the same multinational with $|\mathcal{X}| = 20$ and $B = 24$ hours.

Table 2 shows the performances of OPLB with the three alternative parameter configurations. The configuration found with uRace outperforms the other two configurations on all test instances, saving 10% – 15% on container and pallet usage.

5. DISCUSSION & CONCLUSION

In this paper, we proposed uRace, an innovative tuning algorithm to solve the offline algorithm configuration problem. The unified nature of uRace ensures more efficient usage of

Table 2: Performance of OPLB with its parameters tuned with uRace, compared with well-known construction heuristics, on a set of ten practical loading problem instances from a well-known multinational.

Algorithm	Average Performance on Test Set			# Best
	Containers	Pallets	Runtime (sec.)	
George and Robinson (1980)	2.0	47.4	28	0
Bischoff et al. (1995)	2.0	47.2	261	0
OPLB Tuned with uRace	1.7	44.2	22	10

previously obtained information, and allows important characteristics of the tuning process, such as the exploration-exploitation tradeoff and the locality of the search to be easily adjustable, therefore making uRace readily applicable to a variety of parameter spaces. Moreover, the applicability of uRace extends to other types of tuning. Table 3 shows the suitability of uRace for four types of tuning on a Likert scale. For example, problem instance tuning of stochastic algorithms is in essence a very similar type of tuning. In uRace, fair comparisons between configurations are enforced by simply applying the configurations to the same problem instances. For problem instance tuning of stochastic algorithms, we may similarly enforce fair comparisons between configurations through the use of common random numbers in the stochastic algorithm by imposing identical, synchronous, random number streams (Law and Kelton, 2000).

Table 3: Suitability of uRace for different types of tuning on a Likert scale.

	Type of Parameterized Algorithm	
	Deterministic	Stochastic
Problem Family Tuning	++	+-
Problem Instance Tuning	-	+

For problem family tuning of stochastic algorithms, there are two simultaneous stochastic processes that influence the tuning. Dependent on the application, we may keep one of the two processes fixed (reducing the type of tuning to either problem instance tuning of stochastic algorithms or problem family tuning of deterministic algorithms) or vary both stochastic processes. In the latter case, uRace should be extended with repeated application of the same parameter configurations to the same problem instance.

Besides algorithms, other parameterized stochastic processes can be tuned similarly with uRace. This makes uRace also suitable for simulation optimization purposes. By viewing one additional time unit of simulation running time as a new measurement, and using fixed period batch runs (and therefore fixed length random number streams), such simulation environments can potentially be tuned very efficiently with uRace, eliminating and generating parameter configurations mid-run, while imposing the same random number streams for newly generated experimental settings.

As far as the practical relevance for ORTEC and its customers goes, uRace can be used to tune the parameters of OPLB, thereby automatically customizing the algorithmic framework for each specific customer. To this end, we require a set of training instances, typically a representative selection of the customer’s historic instances. In addition, uRace may be used to periodically update the parameter configuration at the customer site, based on evaluation of more recent, more representative historic problem instances. In other words, the parameter configuration, and therefore the loading algorithm would automatically adjust to the most recent developments at

the customer site.

For future research, it would be interesting to compare the performance of OPLB in combination with uRace with the state-of-the-art loading algorithms discussed by Bortfeldt and Wäscher (2013). To this end, the popularly used set of benchmark instances proposed by Bischoff and Ratcliff (1995) can be used, on which the performance of many existing loading algorithms are reported in the literature. Moreover, through a sensitivity analysis on the input parameters of uRace, their exact effects and their optimal settings in various applications may be exposed.

Finally, we mention two possible extensions of uRace that are specific to problem family tuning of algorithms. First, an extension of the offline algorithm configuration problem of Equation 2, such that we seek for multiple, complementary parameter configurations on the training set, which are subsequently all evaluated on the online problem. Through such an approach, we may be able to improve the quality as well as the robustness of the solutions on the online problem. Second, an extension of the offline tuning effort, from which we take a small selection of well-performing configurations, with an additional online tuning effort, where we allow the parameter configuration (i.e., the construction heuristic) to change during the solution process, and seek for the optimal synergy between the construction heuristics.

ACKNOWLEDGMENTS

Our gratitude goes out to ORTEC for allowing us to conduct an empirical evaluation of the performance of uRace through application to one of their customer's loading problem instances.

References

- Balaprakash, P., Birattari, M., Stützle, T., 2007. Improvement strategies for the f-race algorithm: Sampling design and iterative refinement. *Lecture Notes in Computer Science* 4771, 108–122.
- Birattari, M., 2009. *Tuning metaheuristics: A machine learning perspective*. 2nd ed., Springer, Berlin.
- Birattari, M., Stützle, T., Paquette, L., Varrentapp, K., 2002. A racing algorithm for configuring metaheuristics, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 11–18.
- Bischoff, E., Janetz, F., Ratcliff, M., 1995. Loading pallets with non-identical items. *European Journal of Operations Research* 84, 681–692.
- Bischoff, E., Ratcliff, M., 1995. Issues in the development of approached to container loading. *Omega* 23, 377–390.
- Bortfeldt, A., Wäscher, G., 2013. Constraints in container loading - a state-of-the-art review. *European Journal of Operations Research* 229, 1–20.
- Chang, H., Fu, M., Hu, J., Marcus, S., 2007. *Simulation-based Algorithms for Markov Decision Process*. Springer, Berlin.
- Frazier, P., Powell, W., Dayanik, S., 2008. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization* 47, 2410–2439.
- Fu, M.C., Glover, F., April, J., 2005. Simulation optimization: a review, new developments, and applications., in: *Proceedings of the 2005 Winter Simulation Conference*, pp. 83–95.
- George, J., Robinson, D., 1980. A heuristic for packing boxes into a container. *Computers & Operations Research* 7, 147–156.
- Hong, L., Nelson, B., 2009. A brief introduction to optimization via simulation, in: *Proceedings of the 2009 Winter Simulation Conference*, pp. 76–85.
- Jones, D., Schonlau, M., Welch, W., 1998. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization* 13, 455–492.
- Law, A.M., Kelton, W.D., 2000. *Simulation Modeling & Analysis*. 3rd ed., McGraw-Hill, Inc, New York.
- Lizotte, D., 2008. *Practical Bayesian Optimization*. Ph.D. thesis. University of Alberta. Edmonton, Canada.
- McKay, M., Conover, W., Beckman, R., 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21, 239–245.
- Mes, M., Powell, W., Frazier, P., 2011. Hierarchical knowledge gradient for sequential sampling. *Journal of Machine Learning Research* 12, 2931–2974.
- Wilcoxon, F., 1945. Individual comparisons by ranking methods. *Biometric Bulletin* 1, 80–83.

Working Papers Beta 2009 - 2014

nr.	Year	Title	Author(s)
453	2014	A Unified Race Algorithm for Offline Parameter Tuning	Tim van Dijk, Martijn Mes, Marco Schutten, Joaquim Gromicho
452	2014	Cost, carbon emissions and modal shift in intermodal network design decisions	Yann Bouchery, Jan Fransoo
451	2014	Transportation Cost and CO2 Emissions in Location Decision Models	Josue C. Vélazquez-Martínez, Jan C. Fransoo, Edgar E. Blanco, Jaime Mora-Vargas
450	2014	Tracebook: A Dynamic Checklist Support System	Shan Nan, Pieter Van Gorp, Hendrikus H.M. Korsten, Richard Vdovjak, Uzay Kaymak
449	2014	Intermodal hinterland network design with multiple actors	Yann Bouchery, Jan Fransoo
448	2014	The Share-a-Ride Problem: People and Parcels Sharing Taxis	Baoxiang Li, Dmitry Krushinsky, Hajo A. Reijers, Tom Van Woensel
447	2014	Stochastic inventory models for a single item at a single location	K.H. van Donselaar, R.A.C.M. Broekmeulen
446	2014	Optimal and heuristic repairable stocking and expediting in a fluctuating demand environment	Joachim Arts, Rob Basten, Geert-Jan van Houtum
445	2014	Connecting inventory control and repair shop control: a differentiated control structure for repairable spare parts	M.A. Driessen, W.D. Rustenburg, G.J. van Houtum, V.C.S. Wiers
444	2014	A survey on design and usage of Software Reference Architectures	Samuil Angelov, Jos Trienekens, Rob Kusters
443	2014	Extending and Adapting the Architecture Tradeoff Analysis Method for the Evaluation of Software Reference Architectures	Samuil Angelov, Jos J.M. Trienekens, Paul Grefen
442	2014	A multimodal network flow problem with product Quality preservation, transshipment, and asset management	Maryam SteadieSeifi, Nico Dellaert, Tom Van Woensel
441	2013	Integrating passenger and freight transportation: Model formulation and insights	Veaceslav Ghilas, Emrah Demir, Tom Van Woensel

- 440 2013 [The Price of Payment Delay](#) K. van der Vliet, M.J. Reindorp, J.C. Fransoo
- 439 2013 [On Characterization of the Core of Lane Covering Games via Dual Solutions](#) Behzad Hezarkhani, Marco Slikker, Tom van Woensel
- 438 2013 [Destocking, the Bullwhip Effect, and the Credit Crisis: Empirical Modeling of Supply Chain Dynamics](#) Maximiliano Udenio, Jan C. Fransoo, Robert Peels
- 437 2013 [Methodological support for business process Redesign in healthcare: a systematic literature review](#) Rob J.B. Vanwersch, Khurram Shahzad, Irene Vanderfeesten, Kris Vanhaecht, Paul Grefen, Liliane Pintelon, Jan Mendling, Geofridus G. Van Merode, Hajo A. Reijers
- 436 2013 [Dynamics and equilibria under incremental Horizontal differentiation on the Salop circle](#) B. Vermeulen, J.A. La Poutré, A.G. de Kok
- 435 2013 [Analyzing Conformance to Clinical Protocols Involving Advanced Synchronizations](#) Hui Yan, Pieter Van Gorp, Uzay Kaymak, Xudong Lu, Richard Vdovjak, Hendriks H.M. Korsten, Huilong Duan
- 434 2013 [Models for Ambulance Planning on the Strategic and the Tactical Level](#) J. Theresia van Essen, Johann L. Hurink, Stefan Nickel, Melanie Reuter
- 433 2013 [Mode Allocation and Scheduling of Inland Container Transportation: A Case-Study in the Netherlands](#) Stefano Fazi, Tom Van Woensel, Jan C. Fransoo
- 432 2013 [Socially responsible transportation and lot sizing: Insights from multiobjective optimization](#) Yann Bouchery, Asma Ghaffari, Zied Jemai, Jan Fransoo
- 431 2013 [Inventory routing for dynamic waste collection](#) Martijn Mes, Marco Schutten, Arturo Pérez Rivera
- 430 2013 [Simulation and Logistics Optimization of an Integrated Emergency Post](#) N.J. Borgman, M.R.K. Mes, I.M.H. Vliegen, E.W. Hans

429	2013	Last Time Buy and Repair Decisions for Spare Parts	S. Behfard, M.C. van der Heijden, A. Al Hanbali, W.H.M. Zijm
428	2013	A Review of Recent Research on Green Road Freight Transportation	Emrah Demir, Tolga Bektas, Gilbert Laporte
427	2013	Typology of Repair Shops for Maintenance Spare Parts	M.A. Driessen, V.C.S. Wiers, G.J. van Houtum, W.D. Rustenburg
426	2013	A value network development model and Implications for innovation and production network management	B. Vermeulen, A.G. de Kok
425	2013	Single Vehicle Routing with Stochastic Demands: Approximate Dynamic Programming	C. Zhang, N.P. Dellaert, L. Zhao, T. Van Woensel, D. Sever
424	2013	Influence of Spillback Effect on Dynamic Shortest Path Problems with Travel-Time-Dependent Network Disruptions	Derya Sever, Nico Dellaert, Tom Van Woensel, Ton de Kok
423	2013	Dynamic Shortest Path Problem with Travel-Time-Dependent Stochastic Disruptions: Hybrid Approximate Dynamic Programming Algorithms with a Clustering Approach	Derya Sever, Lei Zhao, Nico Dellaert, Tom Van Woensel, Ton de Kok
422	2013	System-oriented inventory models for spare parts	R.J.I. Basten, G.J. van Houtum
421	2013	Lost Sales Inventory Models with Batch Ordering And Handling Costs	T. Van Woensel, N. Erkip, A. Curseu, J.C. Fransoo
420	2013	Response speed and the bullwhip	Maximiliano Udenio, Jan C. Fransoo, Eleni Vatamidou, Nico Dellaert
419	2013	Anticipatory Routing of Police Helicopters	Rick van Urk, Martijn R.K. Mes, Erwin W. Hans
418	2013	Supply Chain Finance: research challenges ahead	Kasper van der Vliet, Matthew J. Reindorp, Jan C. Fransoo

417	2013	Improving the Performance of Sorter Systems By Scheduling Inbound Containers	S.W.A. Haneyah, J.M.J. Schutten, K. Fikse
416	2013	Regional logistics land allocation policies: Stimulating spatial concentration of logistics firms	Frank P. van den Heuvel, Peter W. de Langen, Karel H. van Donselaar, Jan C. Fransoo
415	2013	The development of measures of process harmonization	Heidi L. Romero, Remco M. Dijkman, Paul W.P.J. Grefen, Arjan van Weele
414	2013	BASE/X. Business Agility through Cross-Organizational Service Engineering	Paul Grefen, Egon Lüftenegger, Eric van der Linden, Caren Weisleder
413	2013	The Time-Dependent Vehicle Routing Problem with Soft Time Windows and Stochastic Travel Times	Duygu Tas, Nico Dellaert, Tom van Woensel, Ton de Kok
412	2013	Clearing the Sky - Understanding SLA Elements in Cloud Computing	Marco Comuzzi, Guus Jacobs, Paul Grefen
411	2013	Approximations for the waiting time distribution In an M/G/c priority queue	A. Al Hanbali, E.M. Alvarez, M.C. van der van der Heijden
410	2013	To co-locate or not? Location decisions and logistics concentration areas	Frank P. van den Heuvel, Karel H. van Donselaar, Rob A.C.M. Broekmeulen, Jan C. Fransoo, Peter W. de Langen
409	2013	The Time-Dependent Pollution-Routing Problem	Anna Franceschetti, Dorothee Honhon, Tom van Woensel, Tolga Bektas, Gilbert Laporte.
408	2013	Scheduling the scheduling task: A time Management perspective on scheduling	J.A. Larco, V. Wiers, J. Fransoo
407	2013	Clustering Clinical Departments for Wards to Achieve a Prespecified Blocking Probability	J. Theresia van Essen, Mark van Houdenhoven, Johann L. Hurink
406	2013	MyPHRMachines: Personal Health Desktops in the Cloud	Pieter Van Gorp, Marco Comuzzi

405	2013	Maximising the Value of Supply Chain Finance	Kasper van der Vliet, Matthew J. Reindorp, Jan C. Fransoo
404	2013	Reaching 50 million nanostores: retail distribution in emerging megacities	Edgar E. Blanco, Jan C. Fransoo
403	2013	A Vehicle Routing Problem with Flexible Time Windows	Duygu Tas, Ola Jabali, Tom van Woensel
402	2012	The Service Dominant Business Model: A Service Focused Conceptualization	Egon Lüftenegger, Marco Comuzzi, Paul Grefen, Caren Weisleder
401	2012	Relationship between freight accessibility and Logistics employment in US counties	Frank P. van den Heuvel, Liliana Rivera, Karel H. van Donselaar, Ad de Jong, Yossi Sheffi, Peter W. de Langen, Jan C. Fransoo
400	2012	A Condition-Based Maintenance Policy for Multi-Component Systems with a High Maintenance Setup Cost	Qiushi Zhu, Hao Peng, Geert-Jan van Houtum
399	2012	A flexible iterative improvement heuristic to Support creation of feasible shift rosters in Self-rostering	E. van der Veen, J.L. Hurink, J.M.J. Schutten, S.T. Uijland
398	2012	Scheduled Service Network Design with Synchronization and Transshipment Constraints For Intermodal Container Transportation Networks	K. Sharypova, T.G. Crainic, T. van Woensel, J.C. Fransoo
397	2012	Destocking, the bullwhip effect, and the credit Crisis: empirical modeling of supply chain Dynamics	Maximiliano Udenio, Jan C. Fransoo, Robert Peels
396	2012	Vehicle routing with restricted loading capacities	J. Gromicho, J.J. van Hoorn, A.L. Kok, J.M.J. Schutten
395	2012	Service differentiation through selective lateral transshipments	E.M. Alvarez, M.C. van der Heijden, I.M.H. Vliegen, W.H.M. Zijm

394	2012	A Generalized Simulation Model of an Integrated Emergency Post	Martijn Mes, Manon Bruens
393	2012	Business Process Technology and the Cloud: Defining a Business Process Cloud Platform	Vasil Stoitsev, Paul Grefen
392	2012	Vehicle Routing with Soft Time Windows and Stochastic Travel Times: A Column Generation And Branch-and-Price Solution Approach	D. Tas, M. Gendreau, N. Dellaert, T. van Woensel, A.G. de Kok
391	2012	Improve OR-Schedule to Reduce Number of Required Beds	J.T. v. Essen, J.M. Bosch, E.W. Hans, M. v. Houdenhoven, J.L. Hurink
390	2012	How does development lead time affect performance over the ramp-up lifecycle?	Andres Pufall, Jan C. Fransoo, Ad de Jong
389	2012	Evidence from the consumer electronics industry	Andreas Pufall, Jan C. Fransoo, Ad de Jong, Ton de Kok
388	2012	The Impact of Product Complexity on Ramp-Up Performance	Frank P.v.d. Heuvel, Peter W.de Langen, Karel H. v. Donselaar, Jan C. Fransoo
387	2012	Co-location synergies: specialized versus diverse logistics concentration areas	Frank P.v.d. Heuvel, Peter W.de Langen, Karel H. v.Donselaar, Jan C. Fransoo
386	2012	Proximity matters: Synergies through co-location of logistics establishments	Frank P. v.d.Heuvel, Peter W.de Langen, Karel H.v. Donselaar, Jan C. Fransoo
385	2012	Spatial concentration and location dynamics in logistics:the case of a Dutch province	Zhiqiang Yan, Remco Dijkman, Paul Grefen
384	2012	FNet: An Index for Advanced Business Process Querying	W.R. Dalinghaus, P.M.E. Van Gorp

383	2012	Defining Various Pathway Terms	Egon Lüftenegger, Paul Grefen, Caren Weisleder
382	2012	The Service Dominant Strategy Canvas: Defining and Visualizing a Service Dominant Strategy through the Traditional Strategic Lens	Stefano Fazi, Tom van Woensel, Jan C. Fransoo
381	2012	A Stochastic Variable Size Bin Packing Problem With Time Constraints	K. Sharypova, T. van Woensel, J.C. Fransoo
380	2012	Coordination and Analysis of Barge Container Hinterland Networks	Frank P. van den Heuvel, Peter W. de Langen, Karel H. van Donselaar, Jan C. Fransoo
379	2012	Proximity matters: Synergies through co-location of logistics establishments	Heidi Romero, Remco Dijkman, Paul Grefen, Arjan van Weele
378	2012	A literature review in process harmonization: a conceptual framework	S.W.A. Haneya, J.M.J. Schutten, P.C. Schuur, W.H.M. Zijm
377	2012	A Generic Material Flow Control Model for Two Different Industries	H.G.H. Tiemessen, M. Fleischmann, G.J. van Houtum, J.A.E.E. van Nunen, E. Pratsini
375	2012	Improving the performance of sorter systems by scheduling inbound containers	Albert Douma, Martijn Mes
374	2012	Strategies for dynamic appointment making by container terminals	Pieter van Gorp, Marco Comuzzi
373	2012	MyPHRMachines: Lifelong Personal Health Records in the Cloud	E.M. Alvarez, M.C. van der Heijden, W.H.M. Zijm
372	2012	Service differentiation in spare parts supply through dedicated stocks	Frank Karsten, Rob Basten
371	2012	Spare parts inventory pooling: how to share the benefits	X.Lin, R.J.I. Basten, A.A. Kranenburg, G.J. van Houtum

	Condition based spare parts supply	
370	2012	Martijn Mes
369	2011	Using Simulation to Assess the Opportunities of Dynamic Waste Collection
		J. Arts, S.D. Flapper, K. Vernooij
368	2011	Aggregate overhaul and supply chain planning for rotatables
		J.T. van Essen, J.L. Hurink, W. Hartholt, B.J. van den Akker
367	2011	Operating Room Rescheduling
		Kristel M.R. Hoen, Tarkan Tan, Jan C. Fransoo, Geert-Jan van Houtum
366	2011	Switching Transport Modes to Meet Voluntary Carbon Emission Targets
		Elisa Alvarez, Matthieu van der Heijden
365	2011	On two-echelon inventory systems with Poisson demand and lost sales
		J.T. van Essen, E.W. Hans, J.L. Hurink, A. Oversberg
364	2011	Minimizing the Waiting Time for Emergency Surgery
		Duygu Tas, Nico Dellaert, Tom van Woensel, Ton de Kok
363	2011	Vehicle Routing Problem with Stochastic Travel Times Including Soft Time Windows and Service Costs
		Erhun Özkan, Geert-Jan van Houtum, Yasemin Serin
362	2011	A New Approximate Evaluation Method for Two-Echelon Inventory Systems with Emergency Shipments
		Said Dabia, El-Ghazali Talbi, Tom Van Woensel, Ton de Kok
361	2011	Approximating Multi-Objective Time-Dependent Optimization Problems
		Said Dabia, Stefan Röpke, Tom Van Woensel, Ton de Kok
360	2011	Branch and Cut and Price for the Time Dependent Vehicle Routing Problem with Time Window
		A.G. Karaarslan, G.P. Kiesmüller, A.G. de Kok
359	2011	Analysis of an Assemble-to-Order System with Different Review Periods
		Ahmad Al Hanbali, Matthieu van der Heijden
358	2011	Interval Availability Analysis of a Two-Echelon, Multi-Item System
		Felipe Caro, Charles J. Corbett, Tarkan Tan, Rob Zuidwijk
357	2011	Carbon-Optimal and Carbon-Neutral Supply Chains
		Sameh Haneyah, Henk Zijm, Marco Schutten, Peter Schuur

356	2011	Generic Planning and Control of Automated Material Handling Systems: Practical Requirements Versus Existing Theory	M. van der Heijden, B. Iskandar
355	2011	Last time buy decisions for products sold under warranty	Frank P. van den Heuvel, Peter W. de Langen, Karel H. van Donselaar, Jan C. Fransoo
354	2011	Spatial concentration and location dynamics in logistics: the case of a Dutch province	Frank P. van den Heuvel, Peter W. de Langen, Karel H. van Donselaar, Jan C. Fransoo
353	2011	Identification of Employment Concentration Areas	Pieter van Gorp, Remco Dijkman
352	2011	BOMN 2.0 Execution Semantics Formalized as Graph Rewrite Rules: extended version	Frank Karsten, Marco Slikker, Geert-Jan van Houtum
351	2011	Resource pooling and cost allocation among independent service providers	E. Lüftenegger, S. Angelov, P. Grefen
350	2011	A Framework for Business Innovation Directions	Remco Dijkman, Irene Vanderfeesten, Hajo A. Reijers
349	2011	The Road to a Business Process Architecture: An Overview of Approaches and their Use	K.M.R. Hoen, T. Tan, J.C. Fransoo G.J. van Houtum
348	2011	Effect of carbon emission regulations on transport mode selection under stochastic demand	Murat Firat, Cor Hurkens
347	2011	An improved MIP-based combinatorial approach for a multi-skill workforce scheduling problem	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
346	2011	An approximate approach for the joint problem of level of repair analysis and spare parts stocking	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
345	2011	Joint optimization of level of repair analysis and spare parts stocks	Ton G. de Kok
344	2011	Inventory control with manufacturing lead time flexibility	Frank Karsten, Marco Slikker, Geert-Jan van Houtum
343	2011	Analysis of resource pooling games via a new extension of the Erlang loss function	Murat Firat, C.A.J. Hurkens, Gerhard J. Woeginger

342	2010	Vehicle refueling with limited resources	Bilge Atasoy, Refik Güllü, Tarkan Tan
341	2010	Optimal Inventory Policies with Non-stationary Supply Disruptions and Advance Supply Information	Kurtulus Baris Öner, Alan Scheller-Wolf Geert-Jan van Houtum
339	2010	Redundancy Optimization for Critical Components in High-Availability Capital Goods	Joachim Arts, Gudrun Kiesmüller
338	2010	Analysis of a two-echelon inventory system with two supply modes	Murat Firat, Gerhard J. Woeginger
335	2010	Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh	Murat Firat, Cor Hurkens
334	2010	Attaining stability in multi-skill workforce scheduling	A.J.M.M. Weijters, J.T.S. Ribeiro
333	2010	Flexible Heuristics Miner (FHM)	P.T. Vanberkel, R.J. Boucherie, E.W. Hans, J.L. Hurink, W.A.M. van Lent, W.H. van Harten
332	2010	An exact approach for relating recovering surgical patient workload to the master surgical schedule	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Nelly Litvak
331	2010	Efficiency evaluation for pooling resources in health care	M.M. Jansen, A.G. de Kok, I.J.B.F. Adan
330	2010	The Effect of Workload Constraints in Mathematical Programming Models for Production Planning	Christian Howard, Ingrid Reijnen, Johan Marklund, Tarkan Tan
329	2010	Using pipeline information in a multi-echelon spare parts inventory system	H.G.H. Tiemessen, G.J. van Houtum
328	2010	Reducing costs of repairable spare parts supply systems via dynamic scheduling	F.P. van den Heuvel, P.W. de Langen, K.H. van Donselaar, J.C. Fransoo
327	2010	Identification of Employment Concentration and Specialization Areas: Theory and Application	Murat Firat, Cor Hurkens

326	2010	A combinatorial approach to multi-skill workforce scheduling	Murat Firat, Cor Hurkens, Alexandre Laugier
325	2010	Stability in multi-skill workforce scheduling	M.A. Driessen, J.J. Arts, G.J. v. Houtum, W.D. Rustenburg, B. Huisman
324	2010	Maintenance spare parts planning and control: A framework for control and agenda for future research	R.J.I. Basten, G.J. van Houtum
323	2010	Near-optimal heuristics to set base stock levels in a two-echelon distribution network	M.C. van der Heijden, E.M. Alvarez, J.M.J. Schutten
322	2010	Inventory reduction in spare part networks by selective throughput time reduction	E.M. Alvarez, M.C. van der Heijden, W.H. Zijm
321	2010	The selective use of emergency shipments for service-contract differentiation	B. Walrave, K. v. Oorschot, A.G.L. Romme
320	2010	Heuristics for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering in the Central Warehouse	Nico Dellaert, Jully Jeunet.
319	2010	Preventing or escaping the suppression mechanism: intervention conditions	R. Seguel, R. Eshuis, P. Grefen.
318	2010	Hospital admission planning to optimize major resources utilization under uncertainty	Tom Van Woensel, Marshall L. Fisher, Jan C. Fransoo.
317	2010	Minimal Protocol Adaptors for Interacting Services	Lydie P.M. Smets, Geert-Jan van Houtum, Fred Langerak.
316	2010	Teaching Retail Operations in Business and Engineering Schools	Pieter van Gorp, Rik Eshuis.
315	2010	Design for Availability: Creating Value for Manufacturers and Customers	Bob Walrave, Kim E. van Oorschot, A. Georges L. Romme
314	2010	Transforming Process Models: executable rewrite rules versus a formalized Java program	S. Dabia, T. van Woensel, A.G. de Kok
313		Getting trapped in the suppression of exploration: A simulation model	

	A Dynamic Programming Approach to Multi-Objective Time-Dependent Capacitated Single Vehicle Routing Problems with Time Windows	
312 2010	Tales of a So(u)rcerer: Optimal Sourcing Decisions Under Alternative Capacitated Suppliers and General Cost Structures	Osman Alp, Tarkan Tan
311 2010	In-store replenishment procedures for perishable inventory in a retail environment with handling costs and storage constraints	R.A.C.M. Broekmeulen, C.H.M. Bakx
310 2010	The state of the art of innovation-driven business models in the financial services industry	E. Lüftenegger, S. Angelov, E. van der Linden, P. Grefen
309 2010	Design of Complex Architectures Using a Three Dimension Approach: the CrossWork Case	R. Seguel, P. Grefen, R. Eshuis
308 2010	Effect of carbon emission regulations on transport mode selection in supply chains	K.M.R. Hoen, T. Tan, J.C. Fransoo, G.J. van Houtum
307 2010	Interaction between intelligent agent strategies for real-time transportation planning	Martijn Mes, Matthieu van der Heijden, Peter Schuur
306 2010	Internal Slackening Scoring Methods	Marco Slikker, Peter Borm, René van den Brink
305 2010	Vehicle Routing with Traffic Congestion and Drivers' Driving and Working Rules	A.L. Kok, E.W. Hans, J.M.J. Schutten, W.H.M. Zijm
304 2010	Practical extensions to the level of repair analysis	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
303 2010	Ocean Container Transport: An Underestimated and Critical Link in Global Supply Chain Performance	Jan C. Fransoo, Chung-Yee Lee
302 2010	Capacity reservation and utilization for a manufacturer with uncertain capacity and demand	Y. Boulaksil; J.C. Fransoo; T. Tan
300 2009	Spare parts inventory pooling games	F.J.P. Karsten; M. Slikker; G.J. van Houtum
299 2009	Capacity flexibility allocation in an outsourced supply chain with reservation	Y. Boulaksil, M. Grunow, J.C. Fransoo
298 2010	An optimal approach for the joint problem of level of repair analysis and spare parts stocking	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
297 2009	Responding to the Lehman Wave: Sales Forecasting and Supply Management during the Credit Crisis	Robert Peels, Maximiliano Udenio, Jan C. Fransoo, Marcel Wolfs, Tom Hendrikx
296 2009	An exact approach for relating recovering surgical patient workload to the master surgical schedule	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Wineke A.M. van Lent, Wim H. van Harten
295 2009	An iterative method for the simultaneous optimization of repair decisions and spare parts stocks	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
294 2009	Fujaba hits the Wall(-e)	Pieter van Gorp, Ruben Jubeh, Bernhard Grusie, Anne Keller

293	2009	Implementation of a Healthcare Process in Four Different Workflow Systems	R.S. Mans, W.M.P. van der Aalst, N.C. Russell, P.J.M. Bakker
292	2009	Business Process Model Repositories - Framework and Survey	Zhiqiang Yan, Remco Dijkman, Paul Grefen
291	2009	Efficient Optimization of the Dual-Index Policy Using Markov Chains	Joachim Arts, Marcel van Vuuren, Gudrun Kiesmuller
290	2009	Hierarchical Knowledge-Gradient for Sequential Sampling	Martijn R.K. Mes; Warren B. Powell; Peter I. Frazier
289	2009	Analyzing combined vehicle routing and break scheduling from a distributed decision making perspective	C.M. Meyer; A.L. Kok; H. Kopfer; J.M.J. Schutten
288	2009	Anticipation of lead time performance in Supply Chain Operations Planning	Michiel Jansen; Ton G. de Kok; Jan C. Fransoo
287	2009	Inventory Models with Lateral Transshipments: A Review	Colin Paterson; Gudrun Kiesmuller; Ruud Teunter; Kevin Glazebrook
286	2009	Efficiency evaluation for pooling resources in health care	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
285	2009	A Survey of Health Care Models that Encompass Multiple Departments	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
284	2009	Supporting Process Control in Business Collaborations	S. Angelov; K. Vidyasankar; J. Vonk; P. Grefen
283	2009	Inventory Control with Partial Batch Ordering	O. Alp; W.T. Huh; T. Tan
282	2009	Translating Safe Petri Nets to Statecharts in a Structure-Preserving Way	R. Eshuis
281	2009	The link between product data model and process model	J.J.C.L. Vogelaar; H.A. Reijers
280	2009	Inventory planning for spare parts networks with delivery time requirements	I.C. Reijnen; T. Tan; G.J. van Houtum
279	2009	Co-Evolution of Demand and Supply under Competition	B. Vermeulen; A.G. de Kok
278	2010	Toward Meso-level Product-Market Network Indices for Strategic Product Selection and (Re)Design Guidelines over the Product Life-Cycle	B. Vermeulen, A.G. de Kok
277	2009	An Efficient Method to Construct Minimal Protocol Adaptors	R. Seguel, R. Eshuis, P. Grefen
276	2009	Coordinating Supply Chains: a Bilevel Programming Approach	Ton G. de Kok, Gabriella Muratore
275	2009	Inventory redistribution for fashion products under demand parameter update	G.P. Kiesmuller, S. Minner
274	2009	Comparing Markov chains: Combining aggregation and precedence relations applied to sets of states	A. Basic, I.M.H. Vliegen, A. Scheller-Wolf
273	2009	Separate tools or tool kits: an exploratory study of engineers' preferences	I.M.H. Vliegen, P.A.M. Kleingeld, G.J. van Houtum

272	2009	An Exact Solution Procedure for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering	Engin Topan, Z. Pelin Bayindir, Tarkan Tan
271	2009	Distributed Decision Making in Combined Vehicle Routing and Break Scheduling	C.M. Meyer, H. Kopfer, A.L. Kok, M. Schutten
270	2009	Dynamic Programming Algorithm for the Vehicle Routing Problem with Time Windows and EC Social Legislation	A.L. Kok, C.M. Meyer, H. Kopfer, J.M.J. Schutten
269	2009	Similarity of Business Process Models: Metrics and Evaluation	Remco Dijkman, Marlon Dumas, Boudewijn van Dongen, Reina Kaarik, Jan Mendling
267	2009	Vehicle routing under time-dependent travel times: the impact of congestion avoidance	A.L. Kok, E.W. Hans, J.M.J. Schutten
266	2009	Restricted dynamic programming: a flexible framework for solving realistic VRPs	J. Gromicho; J.J. van Hoorn; A.L. Kok; J.M.J. Schutten;

Working Papers published before 2009 see: <http://beta.ieis.tue.nl>