

# Evaluating the quality of online optimization algorithms by discrete event simulation

Fabian Dunke<sup>1</sup> · Stefan Nickel<sup>1</sup>

Published online: 14 October 2016  
© Springer-Verlag Berlin Heidelberg 2016

**Abstract** A key feature of dynamic problems which offer degrees of freedom to the decision maker is the necessity for a goal-oriented decision making routine which is employed every time the logic of the system requires a decision. In this paper, we look at optimization procedures which appear as subroutines in dynamic problems and show how discrete event simulation can be used to assess the quality of algorithms: after establishing a general link between online optimization and discrete event systems, we address performance measurement in dynamic settings and derive a corresponding tool kit. We then analyze several control strategies using the methodologies discussed previously in two real world examples of discrete event simulation models: a manual order picking system and a pickup and delivery service.

**Keywords** Online optimization · Algorithm analysis · Discrete event simulation · Order picking · Pickup and delivery

## 1 Introduction

A large variety of industrial applications on the tactical and operational level are characterized by a sequential release of input information over time and the related need for sequential decision making throughout the time horizon (Grötschel and Krumke 2001; Stadtler and Kilger 2008). Online optimization deals with sequential decision

---

✉ Fabian Dunke  
fabian.dunke@kit.edu  
Stefan Nickel  
stefan.nickel@kit.edu

<sup>1</sup> Institute of Operations Research, Discrete Optimization and Logistics, Karlsruhe Institute of Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany

making under incomplete information where each decision must be made on the basis of a limited overseen amount of future input data (lookahead information). Especially over the last two decades, incorporating available data about the near future has become increasingly prominent, driven by electronic data interchange, geographical position systems or intelligent sensor-actuator systems (Ghiani et al. 2004; Psaraftis 1995; Stadler and Kilger 2008). In practice, the task of solving online optimization problems is embedded into a mesh of external processes which are outside the decision maker's control. However, how to solve the arising instances of an online optimization problem is at the decision maker's disposal, and hence, this activity is a crucial recurring pattern in industrial applications. For each decision, an online algorithm is called as a subroutine. It has to determine partial solutions based on the currently available data such that the overall solution at the end of the time horizon will be as good as possible. We are interested in distinguishing good control strategies from bad ones as well as in explaining why algorithms exhibit a certain behavior under a given degree of information. A standard tool to assess the quality of online algorithms is competitive analysis (Karlin et al. 1988; Sleator and Tarjan 1985). Informally, an online algorithm is  $c$ -competitive if for every input it fares at most  $c$ -times as bad as an optimal (omniscient) offline algorithm. A formal definition along with the downsides of this performance measure in real world applications as well as viable alternatives for practice are given in Sect. 3.1.

In this paper, we consider discrete online optimization problems where decisions can be traced back to a discrete structure (Grötschel et al. 2001). These problems occur in different application domains (Borodin and El-Yaniv 1998; Fiat and Woeginger 1998; Grötschel and Krumke 2001) including production and logistics, telecommunications, memory management, self-organizing data structures, or financial engineering. In the application part, we will have a detailed look at two real world example applications where decisions are required from online optimization algorithms repetitively. The first application in Sect. 4.1 considers a manual order picking system in a warehouse where pickers have to retrieve orders from their storage locations throughout the day. The second application in Sect. 4.2 examines a pickup and delivery service in an urban road network. For both applications, a purely analytical approach for determining the quality of different algorithms is out of scope since a high degree of complexity is inherent to the systems as a consequence of the large amount of unknown data and random events. Mathematically, this amounts to a large number of dependent random variables in a discrete event system (Cassandras and Lafortune 2008). Discrete event simulation represents a promising approach to obtain the values of performance indicators in real world applications which are impossible to elicit by exact analysis. The rest of the paper is organized as follows: Sect. 2 lays the methodological foundation for the analysis of online optimization algorithms by utilizing discrete event simulation. Section 3 summarizes performance measurement methods for algorithm quality from the literature and addresses a comprehensive approach for displaying overall algorithm quality. In Sect. 4, we return to the simulation applications and explain algorithm behavior for different amounts of lookahead. For the results, we also make use of the performance measurement approaches presented before. The paper closes with some conclusions in Sect. 5.

## 2 Online optimization and discrete event simulation

In order to justify the adoption of discrete event simulation models for the quality assessment of online optimization algorithms, we discuss analogies between online optimization and discrete event systems.

Let an event in a dynamic system be a spontaneous occurrence triggered by some external entity (event generator, nature) or by fulfillment of all conditions of a switching rule (event generation rule). Because events can be used to submit changes of the input to the system, we obtain the definition of a discrete event system as an event-driven system whose state transitions only depend on the occurrence of discrete events over time (Cassandras and Lafortune 2008). In online optimization, the release of input elements—so-called requests (Grötschel et al. 2001)—occurs at discrete points in time and an online algorithm operates as part of a reactive planning system. Hence, the solution process in online optimization can be modeled as a discrete event system. Because we associate an objective value to a request sequence processed by an optimization algorithm, we include a state valuation function for the objective value of the current state in the formal description of a discrete event system (Cassandras and Lafortune 2008). Table 1 summarizes analogies between the solution procedure in online optimization and the behavior of discrete event systems.

As a result of their similarities, both online optimization and discrete event systems can be tackled with the same arsenal of analytical methods such as automata or Markov chains. However, due to the complexity of real world applications, the computational burden to apply these methods would be too hard as a result of the exploding state space size with increasing level of detail. As a consequence, we have to proceed to another approach which facilitates the determination of quality indicators for the algorithm candidates.

According to Verein Deutscher Ingenieure (1996), “simulation is the reproduction of a system along with its dynamic processes in an executable model in order to retrieve results which are transferable to reality”. Results on system or algorithm performance are derived from first collecting data over a sufficient number of simulation runs which then serve as the computational basis so as to determine (point, interval or distributional) estimates for performance measures of interest

**Table 1** Analogies between online optimization and discrete event systems

Online optimization	Discrete event system
Request	Event
Request sequence	Event sequence
Request arrival	Event occurrence
Request arrivals occur at discrete times	Events occur at discrete times
Algorithm reacts to request arrivals	System reacts to event occurrence
Online algorithm	State transition function
Objective value	State valuation

(Cassandras and Lafortune 2008). Whenever a real world system must be analyzed, but exact analysis is out of reach due to a high degree of complexity and a large number of dependent stochastic processes, simulation is an appropriate tool to provide estimates for desired performance measures. Although the approximation character in the outcomes may be discouraging, there are many advantages: First, simulation may be the only way to analyze complex systems with multiple dependent random variables and manifold types of events such as input element releases, machine breakdowns, occurrence of erroneous data or erroneous operations. Second, sufficiently many simulation runs allow to obtain statistically sound results. Third, simulation allows to track manifold quantities of interest as intended by multicriteria optimization. And fourth, simulation as an abstract model of reality is comparatively risk-free and cheap.

Simulation and optimization can be linked in two hierarchical ways (März and Krug 2011): First, a superordinate optimization procedure may aim at determining optimal parameter values for elements of the system, and hence use a subordinate simulation model just to check the quality of a proposed parameter value. Thus, the simulation evaluates the quality of the solution provided by the optimization and returns the quality value to the optimization method where a new parameter proposal may be generated. This hierarchical relation is used to tune the parameters of some machine or production process on a tactical or strategic planning level. Second, a superordinate simulation model may call a subordinate optimization method iteratively whenever the functional logic of the simulation requires a decision. Clearly, the need for a decision is mainly provoked by new input data. This hierarchical relation is used to emulate the operational behavior of algorithm candidates on real world data in order to come to a conclusion about their quality. To obtain the desired quantities of interest, we generate a sufficient number of sample paths of the state trajectory of the dynamic system by running a sufficient number of independent simulation replications (Cassandras and Lafortune 2008).

At this point, observe the difference between a discrete event system and a discrete event simulation. With a discrete event system we can model online optimization problems. A discrete event simulation finally feeds a discrete event system with input in the form of events over time and advances through its functional logic. Typically, these events comprise classical request arrivals from online optimization, but also all other random events that one would like to model. There are numerous simulation tools available such as Anylogic or Plant Simulation allowing to conduct simulation studies along with graphical output.

Real world systems are inherently complex and contain optimization problems exhibiting an online character. Thus, discrete event simulation is the approximation method of choice as justified by the relationship between online optimization and discrete event systems. Hence, discrete event simulation combined with methods from online optimization represents a powerful tool for systems analysis as it allows to evaluate algorithm performance and identify the most promising algorithm candidates. Disadvantageously, it is much more difficult to gain structural insights on why a system behaves the way it does from simulation experiments than from exact analysis methods. After the simulation model is finished, it is often just used as a black box to evaluate sample state trajectories of the system. For this reason, we also conducted research on exact approaches in a number of elementary problem settings (Dunke 2014); the core

reasons for observed effects can be found in this type of analysis and, as will be seen later, these results can conditionally be transferred to more complex settings.

### 3 Performance measurement in online optimization

We first give a systematic overview of important performance measures found in the literature and then present a more comprehensive approach devised in the first author's thesis [Dunke \(2014\)](#). Formally, the input data corresponds to a sequence  $(\sigma_1, \sigma_2, \dots) \in \Sigma$  of input elements  $\sigma_i$  with  $i \in \mathbb{N}$  where  $\Sigma$  is the set of all possible input sequences. We denote an optimal offline algorithm which knows  $\sigma$  in advance by OPT; the cost of an algorithm ALG on input sequence  $\sigma$  is denoted by  $\text{ALG}[\sigma]$ . The discussion is restricted to minimization problems.

#### 3.1 Literature review

The presentation in this section is subdivided into reviews of deterministic worst-case measures, probabilistic worst-case measures, average-case measures, and distributional approaches.

##### 3.1.1 Deterministic worst-case measures

*Competitive analysis* ([Karlin et al. 1988](#); [Sleator and Tarjan 1985](#)) has become the standard for measuring online algorithm performance. ALG is called  $c$ -competitive if there is a constant  $a$  such that  $\text{ALG}[\sigma] \leq c \cdot \text{OPT}[\sigma] + a$  for all  $\sigma \in \Sigma$ . The competitive ratio  $c_r$  of ALG is the greatest lower bound over all  $c$  such that ALG is  $c$ -competitive. The competitive ratio states how much ALG deviates from OPT due to missing information in the worst case. The measure might fail in several aspects with respect to practice: First, results are overly pessimistic because worst-case instances are decisive. Second, performance is reduced to a single number making it impossible to discriminate between algorithms with equivalent worst-case behavior. Third, competing with an omniscient offline algorithm may be irrelevant in practice. Fourth, competitive analysis cannot be applied that easily to many real world problems due to the complicated nature of those problems and related algorithms.

Several modifications of the competitive ratio were introduced to overcome these weaknesses: Under *resource augmentation* ([Csirik and Woeginger 2002](#); [Kalyanasundaram and Pruhs 2000](#)), the online algorithm is given an extra amount of resources for processing the input sequence. In the *fair adversary* concept ([Blom et al. 2000](#); [Krumke et al. 2002](#)), the offline adversary is restricted to behave in a reasonable way given the request prefix or configuration seen so far. *Loose competitiveness* has been introduced in [Young \(1994\)](#) because worst-case instances are often tailored to problem parameters and also because it is arguable whether low cost inputs should be considered due to setup and overhead. Therefore, inputs leading to small costs are excluded. *Cooperative analysis* ([Dorrigiv and Lopez-Ortiz 2007](#); [Dorrigiv and López-Ortiz 2008](#)) reduces the impact of badly behaving input sequences since these are assumed to be isolated in the input space. Therefore, a "badness" is introduced for each instance and the perfor-

mance guarantee is sought against the badness instead of OPT. The *accommodating function* (Boyar et al. 2002, 2003) is defined for problems with a limited resource which comes in a default amount, but may be varied. The idea then is to evaluate the performance of an algorithm based on those instances where a given amount of the resource is good enough for OPT to fully grant all input elements. *Comparative analysis* (Koutsoupias and Papadimitriou 2000) relates the best objective value of a class of algorithm candidates to that of another class of algorithm candidates which are weaker than OPT, but stronger than those in the first class. The *max/max ratio* (Ben-David and Borodin 1994) compares the (amortized) behavior of algorithms on their respective worst-case instances of a given length. The *relative worst-order ratio* (Boyar and Favrholt 2007) combines the ideas of considering the behavior on input sequence permutations and selecting the worst-case sequence for either algorithm.

### 3.1.2 Probabilistic worst-case measures

Probabilistic worst-case analysis is based on some randomization mechanism. In contrast to average-case analysis, randomness is not used to impose probabilities for input sequences, but to blur worst-case instances. *Smoothed competitive analysis* (Becchetti et al. 2006) originates from smoothed (complexity) analysis (Spielman and Teng 2004). The basic idea is to show that the worst-case character of worst-case inputs vanishes under small (smoothing) perturbations. This has been transferred to algorithm analysis by slightly perturbing instances according to some probability distribution and analyzing the expected competitive ratio on the perturbed sequences. The *random-order ratio* (Kenyon 1996) considers the neighborhood of an input sequence as the set of all its permutations; input instances with length tending to infinity and all permutations of an input sequence are considered equally likely.

### 3.1.3 Average-case measures

In average-case analysis, stochasticity refers to probabilities for input sequences. Competitiveness has been transferred to stochastic settings (Coffman et al. 1980; Scharbrodt et al. 2006). Let  $D$  be a probability distribution over all input sequences, then ALG is called  $c$ -competitive under  $D$  if there is a constant  $a$  such that  $\mathbb{E}_D(\text{ALG}[\sigma]) \leq c \cdot \mathbb{E}_D(\text{OPT}[\sigma]) + a$ . The *expected competitive ratio* of ALG under  $D$  is  $c_r^D = \inf\{c \geq 1 \mid \text{ALG is } c\text{-competitive under } D\}$ . In the same way, the expectation can be taken instance-wise over all ratios: The *expected performance ratio* of ALG under  $D$  is  $c_r^D = \inf\left\{c \geq 1 \mid \mathbb{E}_D\left(\frac{\text{ALG}[\sigma]}{\text{OPT}[\sigma]}\right) \leq c\right\}$ . The *diffuse adversary model* (Koutsoupias and Papadimitriou 2000) decreases adversary power by restricting instances to a class of distributions  $\Delta$  whereof a worst distribution  $D \in \Delta$  is selected. In the *statistical adversary model* Raghavan (1991), inputs have to be generated according to given statistical assumptions. In paging with locality of reference, the *Markov model* of Karlin and Phillips (2000) extends access graphs with edge probabilities such that input sequences are generated by a Markov chain. Franaszek and Wagner (1974) considered the expected number of page faults per unit time in a *distributional model*

of paging where pages are generated according to an arbitrary distribution one after another.

### 3.1.4 Distributional measures

The main advantage of distributional performance analysis is that an algorithm is judged by distributional information instead of a single indicator. *Relative interval analysis* (Dorrigiv et al. 2009) only considers the extreme values of a distribution for two algorithms. The relative interval of an algorithm pair corresponds to its asymptotic range of costs per input element. *Stochastic dominance* (Müller and Stoyan 2002) establishes an order relation between distributions of random variables. By interpreting the objective value as a random variable, this concept has been transferred to online optimization (Hiller 2009). Let  $F_{\text{ALG}} : \mathbb{R} \rightarrow [0, 1]$  be the cumulative distribution function of the objective value of ALG. Then  $\text{ALG}_1$  dominates  $\text{ALG}_2$  stochastically if and only if  $F_{\text{ALG}_2}(v) \geq F_{\text{ALG}_1}(v)$  for all  $v \in \mathbb{R}$ . *Bijective analysis* (Angelopoulos et al. 2007) is a special case of stochastic dominance where input sequences are uniformly distributed.

## 3.2 Comprehensive performance measurement

First observe that none of the previous approaches other than competitive analysis has earned broad acceptance, probably because of their involved definitions which impedes their general applicability. We discuss the performance measurement approach devised in the first author's thesis (Dunke 2014) which summarizes the global behavior of an algorithm over all instances and also takes into account local (instancewise) quality. The approach admits an image of algorithm quality that is free of any risk preference. In the following,  $f$  is a function returning the objective value for any pair  $(\sigma, s(\sigma))$  consisting of an input sequence  $\sigma$  and the solution  $s(\sigma)$  that is produced by algorithm ALG upon processing  $\sigma$ .

**Definition 3.1** (*Objective value*) Let  $\sigma \in \Sigma$  be an input sequence, let ALG be an algorithm choosing solution  $s_{\text{ALG}}(\sigma)$  on  $\sigma$ , and let  $f$  be a real-valued function evaluating the pair  $(\sigma, s_{\text{ALG}}(\sigma))$ . Then  $v_{\text{ALG}}(\sigma) := f(\sigma, s_{\text{ALG}}(\sigma))$  is called the *objective value* of ALG with respect to  $\sigma$ .

Since online optimization generally assumes that no probabilities for instance occurrences are given, the maximum entropy distribution emulates the state of a complete lack of information best (Jaynes 1957; Jaynes 1957). The uniform distribution over  $\Sigma$  is the maximum entropy distribution with support  $\Sigma$ . For finite  $\Sigma$ , this leads to counting results saying how many input sequences out of all of them yield a certain objective value (Hiller 2009). The counting distribution function subsumes this frequency information:

**Definition 3.2** (*Counting distribution function of objective value*) Let  $\sigma \in \Sigma$  be an input sequence, let ALG be an algorithm, and let  $v_{\text{ALG}}(\sigma)$  be the objective value of ALG on  $\sigma$ . If  $\Sigma$  is a discrete set, then the function  $F_{\text{ALG}} : \mathbb{R} \rightarrow [0, 1]$  with

$$F_{\text{ALG}}(v) := \frac{\sum_{\sigma \in \Sigma} \mathbf{1}_{(-\infty, v]}(v_{\text{ALG}}(\sigma))}{|\Sigma|}$$

is called the *counting distribution function of the objective value* of ALG over  $\Sigma$ , where the indicator function  $\mathbf{1}_{(-\infty, v]}(v_{\text{ALG}}(\sigma))$  is 1 if  $v_{\text{ALG}}(\sigma) \in (-\infty, v]$  and 0 otherwise.

The following definitions account for the relative performance of two algorithms to each other when both are restricted to operate on the same problem instance:

**Definition 3.3** (*Performance ratio*) Let  $\sigma \in \Sigma$  be an input sequence, let  $\text{ALG}_1, \text{ALG}_2$  be two algorithms choosing solutions  $s_{\text{ALG}_1}(\sigma), s_{\text{ALG}_2}(\sigma)$  on  $\sigma$ , respectively, and let  $f$  be a real-valued function evaluating the pair  $(\sigma, s_{\text{ALG}}(\sigma))$ . Then  $r_{\text{ALG}_1, \text{ALG}_2}(\sigma) := \frac{f(\sigma, s_{\text{ALG}_1}(\sigma))}{f(\sigma, s_{\text{ALG}_2}(\sigma))}$  is called the *performance ratio* of  $\text{ALG}_1$  relative to  $\text{ALG}_2$  with respect to  $\sigma$ .

The maximum  $r_{\text{ALG}_1, \text{ALG}_2}(\sigma)$  over all  $\sigma \in \Sigma$  coincides with the competitive ratio for an online optimization problem, if  $\text{ALG}_1$  is an online algorithm and  $\text{ALG}_2$  equals OPT.

**Definition 3.4** (*Counting distribution function of performance ratio*) Let  $\sigma \in \Sigma$  be an input sequence, let ALG be an algorithm, and let  $r_{\text{ALG}_1, \text{ALG}_2}(\sigma)$  be the performance ratio of  $\text{ALG}_1$  relative to  $\text{ALG}_2$  on  $\sigma$ . If  $\Sigma$  is a discrete set, then the function  $F_{\text{ALG}_1, \text{ALG}_2} : \mathbb{R} \rightarrow [0, 1]$  with

$$F_{\text{ALG}_1, \text{ALG}_2}(r) := \frac{\sum_{\sigma \in \Sigma} \mathbf{1}_{(-\infty, r]}(r_{\text{ALG}_1, \text{ALG}_2}(\sigma))}{|\Sigma|}$$

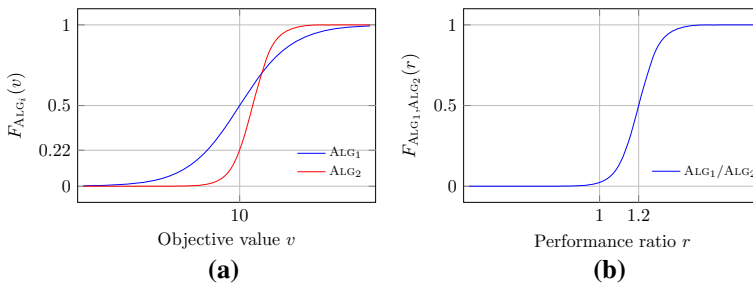
is called the *counting distribution function of the performance ratio* of  $\text{ALG}_1$  relative to  $\text{ALG}_2$  over  $\Sigma$ , where the indicator function  $\mathbf{1}_{(-\infty, r]}(r_{\text{ALG}_1, \text{ALG}_2}(\sigma))$  is 1 if  $r_{\text{ALG}_1, \text{ALG}_2}(\sigma) \in (-\infty, r]$  and 0 otherwise.

The numerical example in Fig. 1 (a) tells us that  $\text{ALG}_1$  attains an objective value of at most 10 on 50% of all input sequences. For  $\text{ALG}_2$ , only 22% of the inputs lead to an objective value of at most 10. In part (b), we see that the largest part of all input sequences leads to a performance ratio larger than 1. Moreover, 50% of all input sequences lead to a ratio smaller than 1.2 and the other 50% have a ratio of at least 1.2.

Distribution functions offer both a global view on algorithm quality over all instances and a local view on algorithm quality relative to some reference algorithm on the same instance. Distribution-based analysis also yields information about ranges and variability. Presenting a decision maker with the counting distribution functions eliminates the burden of defining trade-offs between worst case, best case and average case: The decision maker is equipped with all information except for the direct mapping of objective values and performance ratios to input instances; we cannot think of a more comprehensive method.

In case of a high number of algorithm alternatives, we recommend a two-staged process of empirical performance assessment: In the first stage, an average-case analysis allows us to find the most promising algorithm candidates in terms of expected



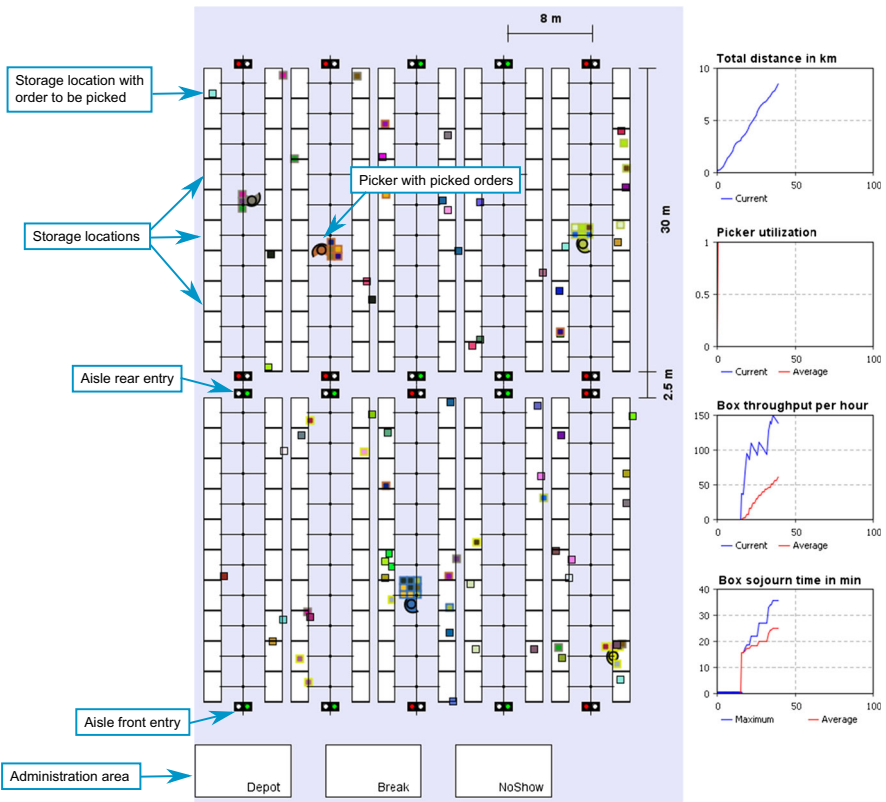


**Fig. 1** Exemplary illustration of counting distribution functions of **a** objective value  $v$  of  $\text{ALG}_1$  and  $\text{ALG}_2$ , and of **b** performance ratio  $r$  of  $\text{ALG}_1$  relative to  $\text{ALG}_2$

algorithm behavior and to filter out those algorithms which shall be investigated further. In the second stage, distributional analysis leads to a fine-grained assessment of each candidate's risk profile with respect to attainable objective values and performance ratios. To support the statistical validity of the sampled results, one has to ensure that a sufficient number of input sequences are considered. Therefore, it is advisable to choose the sample size large enough to ensure a confidence interval (to a given confidence level) of reasonable width for the mean objective value and performance ratio (Cassandras and Lafortune 2008). With this sample size, the distributional analysis is likely to display a proper image of the variability in an algorithm's outcome. However, it is undisputed that isolated worst-case instances cannot be detected by such an empirical approach.

## 4 Applications

We return to the two applications outlined in Sect. 1. Due to the high complexity incurred by many dependent random variables and multiple optimization goals, we use simulation to analyze algorithm performance. The goal of the study is to answer the question “How much can the solution quality be improved by providing information at an earlier point in time by which type of algorithm candidates?”. This question is motivated by recent advances in information and communications technology, such as electronic data interchange or geographical position systems, which allow to access information earlier. Accordingly, we define lookahead as a mechanism which makes information known earlier by  $D$  time units (lookahead duration) compared to the no-lookahead case. In Dunke (2014) it is shown that in the classical traveling salesman problem (TSP) large tour length reductions can be achieved by lookahead through improved resequencing possibilities and avoiding detours. We will resort to this result to explain algorithm behavior in the simulation models. Simulation models were developed in AnyLogic 6.9.0; algorithms were implemented in the Java environment of AnyLogic and MIP formulations were solved using IBM ILOG CPLEX 12.5 with a time limit of 120 seconds. Experiments were performed on a machine with AMD Phenom II X6 1100T 3.31 GHz processor and 16 GB RAM under Microsoft Windows 7 (64-bit). Detailed result data (including average results,



**Fig. 2** Animation of the simulation model for an order picking system

distributional results, and 95 % confidence intervals) is available online at <http://dolor.kit.edu/english/downloads.php>.

#### 4.1 Manual order picking in warehouses

In a manual order picking system (see Fig. 2), pickers move through the aisles of a warehouse to retrieve items packed in boxes as demanded by the orders from external customers. When a picker has collected all boxes assigned to him, he returns to a depot to unload the boxes and wait for the next assignment of boxes to be picked. Typically, customer orders arrive throughout the day and the objective is to make the pickers collect all boxes of the customer orders in a way that meets the decision maker's goal system best. The following quality indicators for the routes are used to judge the quality of the responsible algorithm: Makespan, total distance covered by all pickers, picker utilization, box throughput. Observe that a realistic order picking system is also subject to several further stochastic events such as blocking effects, or individual picker behavior.

The layout consists of ten aisles, each with 20 storage locations, arranged in a lower and an upper block. The depot and break location are positioned in the lower

left corner. The aisle length amounts to 30 m, the horizontal (vertical) aisle distance is 8 (2.5) m (cf. Fig. 2). On average, pickers move at a speed of 1 m per second in order to fulfill the orders according to the pick lists that are given to them at the depot. Aisle traversal is subject to blocking effects (Huber 2011), e.g., because of security or space considerations: Only one picker is allowed inside an aisle at a time. Five pickers have to serve  $n = 625$  orders over a work day of 600 min plus potential overtime. An order may consist of up to three boxes and the picker capacity amounts to ten boxes. Orders have to be brought to the depot where they are further processed for distribution. Order arrival and data are random, i.e., release time, number of boxes, pick time, drop time and box locations are realizations of random variables that are unknown to the algorithms which have to determine pick lists and routes. In addition, we have the following random influences: Picker velocity profile, picker break start and end times, picker no-show occurrence, and if applicable, picker no-show start and end times. After execution of a simulation replication, we obtain the makespan, the total distance covered by all pickers, the picker utilization and the box throughput as quality indicators for computed pick lists and routes. Lookahead appears as time lookahead of duration  $D \in \{0, 60, 120, \dots, 600\}$  min, i.e., time windows of the next  $D$  minutes are seen at each time. Once an order arrives, it is ready to be picked, i.e., orders can be picked earlier through lookahead. We draw 50 independent simulation replications. The selected parameter setting may represent the warehousing operations of a producer which produces homogeneous goods in a medium volume.

*Algorithms.* An algorithm is required to determine the pick lists and routes for all pickers available at that time (see, e.g., the survey in Henn et al. (2012)). Sequential methods cope with the complexity of the problem by decoupling the problems: A batching algorithm assigns orders to pickers; a routing algorithm determines picker traversal paths through the aisles including aisle entry and exit points. Simultaneous methods solve the batching and routing problem at the same time. However, when many customer orders are known, e.g., due to large lookahead, problem instances become too huge to be solved by a simultaneous method.

#### *Batching Algorithms*

- **PRIORITYBATCHING** (PRIO, Henn et al. (2012)): Sort orders according to a criterion, e.g., non-increasingly according to their number of boxes. Assign orders successively to batches in a first fit manner.
- **SEEDBATCHING** (SEED, Henn et al. (2012)): Batches are built sequentially: Initialize each batch with a seed order, e.g., by selecting an order with the largest number of boxes; fill the batch with additional orders according to an order congruency rule, e.g., select an order with the smallest number of additional aisles.
- **SAVINGSBATCHING** (SVGS, Henn et al. (2012)): Batches are built simultaneously: Initialize the batch building process with each order forming a separate batch. In the improvement phase, combine orders of two batches into one batch if the total distance is reduced according to the routing algorithm applied until no further improvement is possible.

- **LOCALSEARCHBATCHING (LS, Henn et al. (2012))**: Let the neighborhood of a batch set be given by all batch sets which are obtained by a swap or shift move: A swap move exchanges one selected order per batch between two batches; a shift move transfers a selected order from one batch to another. A perturbation of a batch set consists of transferring a random number of orders from one batch to another if the receiving batch remains feasible. Execute the following two steps until the total distance of all batches cannot be reduced in either step: Search within the neighborhood of the current batch set for a batch set with smaller total distance. Perturb the current batch set for a fixed number of times to find a batch set with smaller total distance.
- **TABUSEARCHBATCHING (TS, Henn et al. (2012))**: Let the neighborhood of a batch set be defined as in **LOCALSEARCHBATCHING**, but with the modification that those swap and shift moves, along with their inverse moves, that have been carried out within a prescribed number of previous iterations are excluded from the neighborhood. Let a perturbation of a batch set be defined as in **LOCALSEARCHBATCHING**. Execute the following two steps until the total distance of all batches cannot be reduced in the second step: For a fixed number of times, select within the neighborhood of the current batch set the batch set with smallest total distance as the new current batch set. Perturb the current batch set for a fixed number of times to find a batch set with smaller total distance.

The last three batching algorithms (SVGS, LS, TS) already take into account potential routes and the resulting sum of route lengths (total distance) as determined by the applied routing method. However, since in the course of algorithm execution the routing algorithm is applied only as a subroutine for evaluating the quality of a given batch, the phases of batch building and batch routing still run successively.

### *Routing Algorithms*

- **RETURNROUTING (RET, Henn et al. (2012))**: Lower aisles with boxes are visited first from left to right; upper aisles with boxes are visited afterwards from right to left. Each aisle except for the last lower aisle with a box is entered and exited at its front entry; the last lower aisle with a box is traversed entirely.
- **S-SHAPEDROUTING (S, Henn et al. (2012))**: First, the two leftmost aisles with boxes of both blocks are traversed upwards entirely. Second, upper aisles with boxes are visited from left to right where each aisle except for the rightmost is traversed entirely in the direction opposite to that of the previous aisle; the rightmost aisle is traversed entirely if it is entered at its rear entry, otherwise it is entered and exited at its front entry. Third, lower aisles with boxes are visited from right to left where each aisle except for the rightmost aisle and the last lower aisle is traversed entirely in the direction opposite to that of the previous aisle; the rightmost aisle is traversed downwards entirely, the last lower aisle is traversed entirely if it is entered at its rear entry, otherwise it is entered and exited at its front entry.
- **LARGESTGAPROUTING (GAP, Henn et al. (2012))**: The largest gap of an aisle is its largest segment that contains no box, i.e., either the segment between two adjacent boxes, between front entry and lowermost box, or between rear entry and uppermost box. The largest gap of an aisle separates two parts of the aisle from

each other: The lower (upper) part starts at the front (rear) entry and finishes at the lowermost (uppermost) point of the largest gap. First, lower parts of lower aisles with boxes are visited from left to right. Second, upper parts of lower aisles with boxes are visited from right to left. Third, lower parts of upper aisles with boxes are visited from left to right. Fourth, upper parts of upper aisles with boxes are visited from right to left. The rightmost aisle of each block is traversed entirely; in all other aisles, lower (upper) parts are entered and exited at the front (rear) entry.

- OPTIMALROUTING (OPT, Lawler et al. (1985), Miller et al. (1960)): Solve a mixed integer programming (MIP) formulation of the order routing problem (which amounts to a TSP instance). Visit the boxes in the order suggested by the obtained solution.

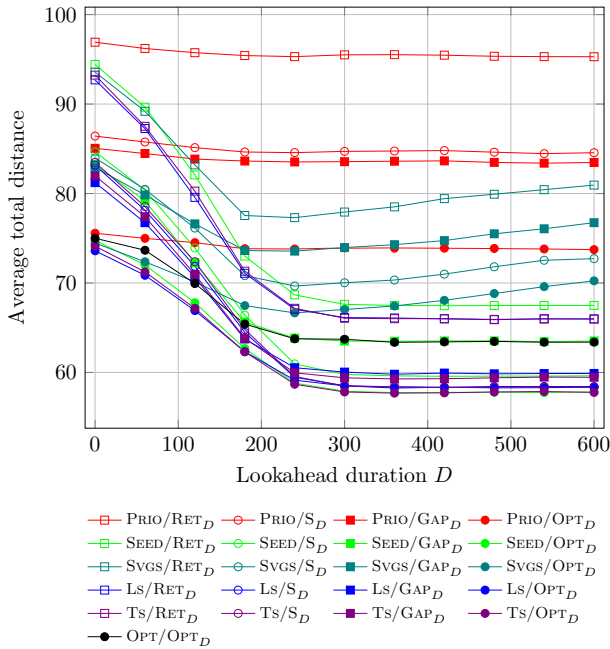
#### *Simultaneous Batching and Routing Algorithm*

- OPTIMAL,OPTIMAL (OPT/OPT, Dunke (2014)): Solve an MIP formulation of the order batching and routing problem. Assign orders to pickers and apply for each picker's boxes the visiting order as suggested by the obtained solution.

**Average Results.** All combinations of batching and routing algorithms as well as the simultaneous algorithm were tested. We note that rule-based routings (RET, S, GAP) are accepted in practice because of their simplicity, whilst optimal routes as calculated by OPT or the simultaneous approach may be shapeless. We restrict attention to the total travel distance in kilometers and the box throughput in boxes per hour. Figure 3 shows that a substantial reduction in the total distance covered by all pickers is achieved by lookahead. For lookahead durations  $D \geq 300$ , optimization potential is satiated because of picker capacities and the sufficient number of known orders to produce "good" pick lists and routes. The effect is explained by the same change of processing restrictions as experienced in the TSP (cf. Dunke (2014)): Since a box may be picked up as soon as it becomes known, increasing the lookahead duration leads to increased probabilities for spatially proximate boxes.

Batching by PRIO suffers poor overall performance and is eliminated from further consideration; SVGS is also rejected as it consistently loses to all remaining batching policies. SEED, LS and TS exhibit comparable behavior. We draw a more precise picture based on the selected routing strategy: RET fails due to its naive approach. Concerning routing policy OPT, all three batching algorithms are considered equal. Under GAP routing and small lookahead, LS has a slight advantage over the two other batching rules, whereas in the case of medium to large lookahead TS excels LS and SEED. Under routing policy S, all algorithms exhibit the same quality; however, performance is degraded as compared to OPT's routing for small lookahead. Simultaneous batching and routing by OPT/OPT is impracticable to obtain solutions quickly: Whenever more than ten orders, i.e., up to 30 boxes, were open, we had to forfeit almost always the exact reoptimization to apply SEED/GAP as a computationally tractable substitute. Thus, OPT/OPT and SEED/GAP coincide for  $D \geq 180$ .

Based on the total distance objective only, we come to the interim conclusion that TS, LS and SEED lead to the most promising batchings along with either S or OPT as recommendable routing policies. Exact reoptimization is computationally too hard. Before we proceed to the box throughput, we note that travel distance and box



**Fig. 3** Average total distances for different lookahead durations and  $n = 625$  in the order picking system

throughput are not competing goals. Thus, we do not expect throughput to be negatively affected by distance minimization.

The box throughput attained by the pickers is consistent with the previously found ranking of batching and routing algorithms based on the total distance (cf. Fig. 4). However, while differences in throughput as accomplished by the algorithms are virtually non-existent for small lookahead durations, batching algorithms TS, LS and SEED unfold their potential and consistently outperform all other batching policies as well as exact reoptimization for larger lookaheads. Routings by S and OPT clearly outperform the other routing strategies.

**Distributional results** Since routing policy OPT intrinsically leads to the shortest route for a given batch, we restrict the discussion to this routing strategy. Because batching algorithm TS has been identified as one of the top candidates, we select TS/OPT<sub>600</sub> as the reference for performance ratios relative to the “best” offline algorithm. We also restrict ourselves to batching algorithm candidates TS, LS and SEED which emerged superior from the average-case analysis. Figure 5 confirms the positive effect of lookahead on the total distance. Concerning batching algorithm candidates TS, LS and SEED, a perfect ordering of the empirical counting distribution functions is observed for successive lookahead durations up to  $D = 240$ . Hence, we have an exclusively beneficial lookahead effect for these information regimes. For  $D \geq 300$ , empirical counting distribution functions intransparently cross each other countless times so that no exclusive benefit can be concluded for these lookahead durations; the marginal benefit of an additional time unit of lookahead is approxi-

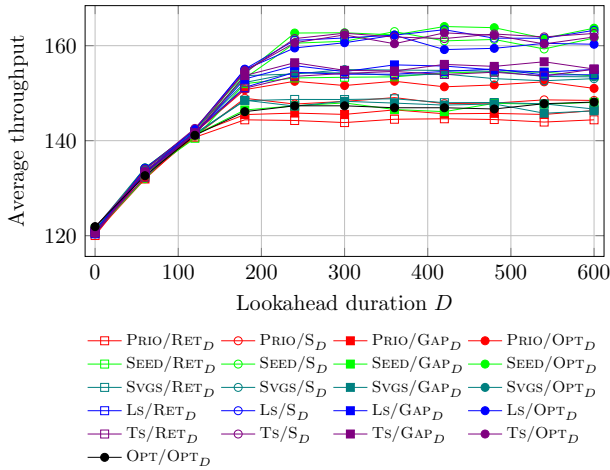


Fig. 4 Average throughput for different lookahead durations and  $n = 625$  in the order picking system

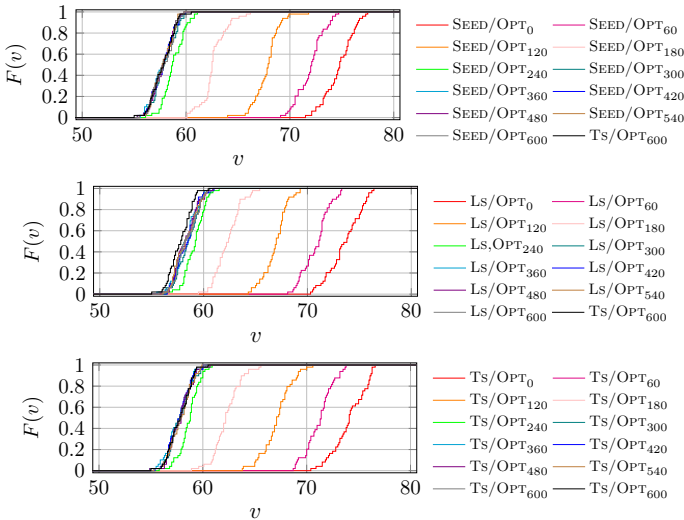
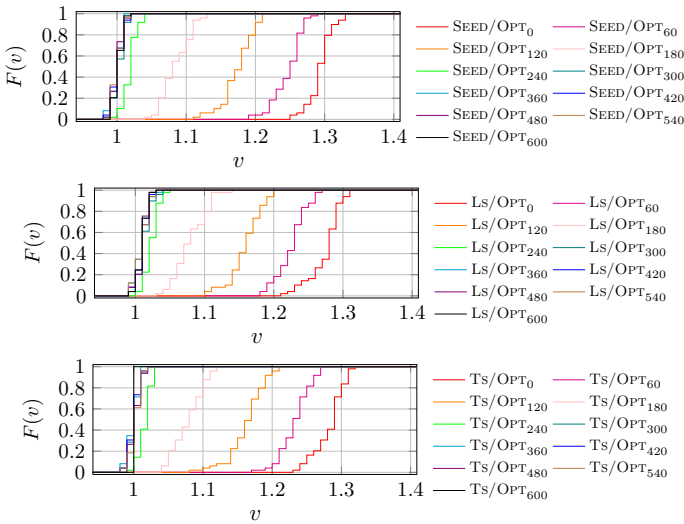


Fig. 5 Empirical counting distribution functions of distance for  $n = 625$  in the order picking system

mately zero. Each plot has large steepness in a characteristic interval of total distance values and nearly no steepness elsewhere indicating that each algorithm (combined with lookahead duration) corresponds to a specific range of total distances. However, for larger lookahead durations there are instances where additional lookahead leads to an objective value degradation; since the marginal benefit of lookahead in this part is negligible, this effect is considered unimportant. The empirical counting distribution functions of the performance ratio for the total distance relative to Ts,OPT $_{600}$  in Fig. 6 show that experimental competitive ratios are not larger than 1.33 for Ts, Ls and SEED. Compared with the “unrestricted” TSP (cf. Dunke (2014)), distance



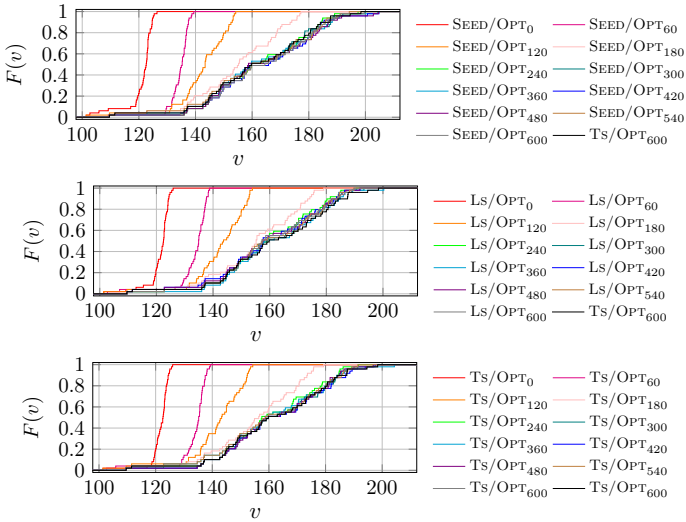
**Fig. 6** Empirical counting distribution functions of performance ratio of distance relative to Ts/OPT<sub>600</sub> for  $n = 625$  in the order picking system

savings are still significant albeit their magnitude is reduced due to picker capacities. The gap between the plots of successive lookahead levels admits the same conclusions as drawn from the total distance distributions. For lookahead  $D \geq 300$ , ratios are centered around 1, i.e., (slight) deterioration despite additional lookahead is seen regularly.

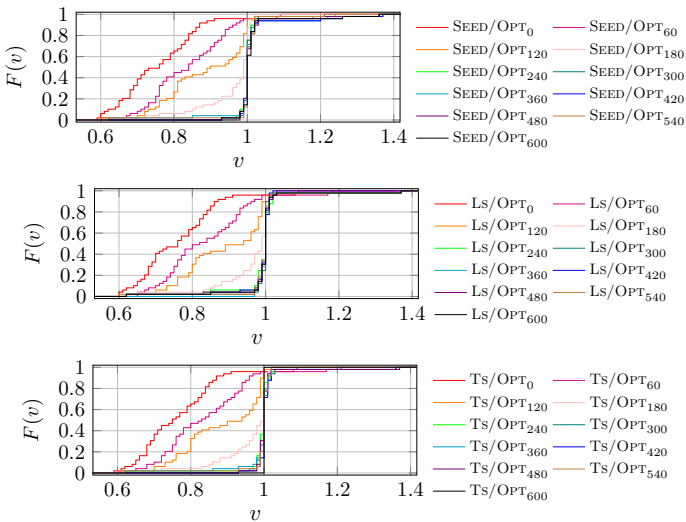
Figures 7 and 8 illustrate the distributional results for the box throughput. Because of maximization, it is desirable to have more mass on larger values. Obviously, lookahead allows pickers to achieve a higher throughput as compared to the online case where generation of full pick lists is impeded and waiting times are more probable. We observe a qualitative difference compared to the total distance case: Variability increases considerably for increasing lookahead durations; likewise, the variability of performance ratios relative to Ts/OPT<sub>600</sub> decreases considerably for increasing lookahead durations (the curves are less steep): In the pure online case, orders arrive over the entire time horizon and the throughput is heavily affected by the arrival process, particularly by the last orders; under full lookahead the throughput is a sole consequence of the picker efficiency in coping with the input sequence revealed at the outset. Hence, throughput is inherently throttled and regulated in the online case leading to more invariant behavior compared to the case where the system evolves freely. For total distance there is no such effect because the distance walked by the pickers is independent of time considerations. In contrast, throughput is measured in boxes per unit time such that the time needed for serving all boxes heavily influences this objective.

Applying the distributional analysis, we were able to delimit the total distance into a range with a width of approximately 8 kilometers irrespective of the lookahead duration. The variability of the total distance is unaffected by the lookahead level. In contrast to that the variability of the throughput increases significantly with additional





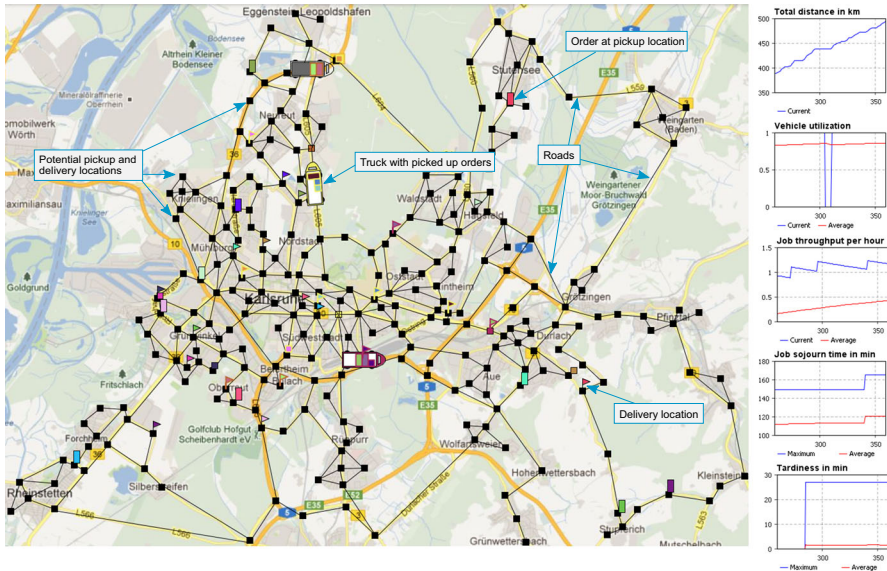
**Fig. 7** Empirical counting distribution functions of throughput for  $n = 625$  in the order picking system



**Fig. 8** Empirical counting distribution functions of performance ratio of throughput relative to  $Ts/OPT_{600}$  for  $n = 625$  in the order picking system

lookahead as explained above. Nonetheless, the main mass of observed throughput values under full lookahead is far higher than those observed in the case of no lookahead.

We conclude this section by pointing out that in manual order picking systems where boxes can be picked up once they are known, massive improvements in all goals could be observed as a result of an enlarged planning basis. Since the objectives are not conflicting, all goals can be improved and no trade-offs between them have to be



**Fig. 9** Animation of the simulation model for a pickup and delivery service

taken into account. From a managerial point of view, we recommend to install technical devices which allow for the retrieval of lookahead information and to check whether operating strategies currently implemented in the warehouse conform to potential warehouse efficiency as extracted by simulation under batching policies TS, LS and SEED combined with routing policies S and OPT.

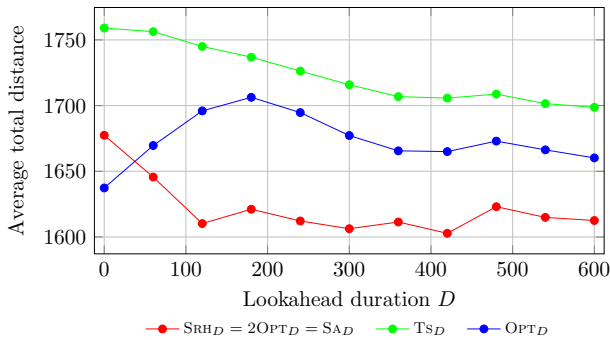
## 4.2 Pickup and delivery service in an urban road network

In a pickup and delivery service (see Fig. 9), customers specify transportation orders between individual origins and destinations in a road network. In addition, customers provide preferred time windows for their pickup and delivery time. Transportation orders are served by a fleet of vehicles and each vehicle has to start and end its routes at an individual depot. Typically, transportation orders arrive throughout the day and the objective is to make the vehicles pick up and deliver all transportation orders in a way that meets the decision maker's goal system best. This problem setting naturally adheres to time-related objectives such as the mean and maximum tardiness over all pickup and delivery orders. Additionally, quality indicators such as the makespan, total distance covered by all vehicles, vehicle utilization, or order throughput are relevant. We observe that in the pickup and delivery service, the decision maker faces conflicting objectives: A prescribed service level with respect to keeping the time windows can conflict with requiring short routes which would result from spatial considerations only. In addition, there are a number of side effects such as traffic jams, varying vehicle speeds, or individual driver behavior.

The network represents the region of the city of Karlsruhe in Germany and consists of 269 central points and 449 major roads between them (cf. Fig. 9). The network diameter is 28.5 kilometers; each road is prescribed a maximum speed limit. Vehicles are subject to the current traffic scene, i.e., traffic jams represent additional random events. Three vehicles have to serve  $n = 50$  customer orders arriving over a work day of 600 min plus potential overtime. An order may consist of up to two units to be transported and the vehicle capacity amounts to five units. Order arrival and data are random, i.e., release time, number of units to be transported, pickup time, delivery time, pickup time window, delivery time window, pickup location and delivery location of an order are realizations of random variables that are unknown to the algorithms which have to determine the routes of the vehicles. In addition, we have the following random influences: Driver break start and end times, driver no-show occurrence, and if applicable, driver no-show start and end times. After execution of a simulation replication, we obtain the makespan, the total distance covered by all vehicles, the mean and maximum tardiness over all pickup and delivery orders, the vehicle utilization and the order throughput as quality indicators for computed vehicle routings. Lookahead appears as time lookahead of duration  $D \in \{0, 60, 120, \dots, 600\}$  min. Once a transportation request arrives, it has to wait until its time window starts before the order can be picked up or delivered, respectively. We draw 100 independent simulation replications. The selected parameter setting may represent the situation of a small home health care provider in a medium-sized urban region.

**Algorithms.** An algorithm is required to determine the routes for all vehicles available at that time so as to fulfill the known and yet unfulfilled transportation orders. In contrast to the order picking system, temporal restrictions have to be regarded, earliest start times for pickups and deliveries are not forwarded through lookahead but retained due to the time windows specified by the customers, and assignment decisions may be revoked. Because of time, not only spatial proximity of locations but also temporal proximity of time windows matters; each pickup has to be seen in logical conjunction with the corresponding delivery operation. Therefore, the variety of solution methods is more limited than for problems without time windows and there is no a-priori subdivision into batching and routing strategies. Some algorithms resort to a performance measure for the quality of a route. We evaluate route quality by aggregating total travel distance, average tardiness and maximum tardiness into an auxiliary objective function in the form of a linear combination (scalarization). Despite extensive numerical tests, there was no coefficient setting which consistently outperformed another one over all three goals. In the end, we weighted the average tardiness (in minutes) with a coefficient of 5, maximum tardiness (in minutes) with a coefficient of 1, and total travel distance (in km) with a coefficient of 0.01. In our opinion, this reflects best what a decision maker could perceive as a fair trade-off between the different goals. Except for the tabu search heuristic and the exact reoptimization approach, all of the following algorithms are modified versions of the algorithms provided by Kallrath (Kallrath 2005) for vehicle routing on hospital campuses. The neighborhood structure in the tabu search algorithm is taken from the setting of dial-a-ride problems discussed by Cordeau and Laporte (Cordeau and Laporte 2003).

- SEQUENCINGREASSIGNMENTHEURISTIC (SRH, part 2 of [Kallrath \(2005\)](#)):
  1. *Assignment of orders to vehicles*: Sort unassigned orders by non-decreasing earliest pickup times and assign orders within a time slice of prescribed length (e.g., 100 min) to vehicles by a modified first fit rule which ensures that orders with close earliest pickup time (e.g., less than 25 min) are not assigned to the same vehicle. Assign previously unassigned orders by the (unmodified) first fit rule.
  2. *Route construction*: Create a route for each vehicle by successively inserting its assigned orders (pickup and delivery location) in a best possible way in terms of a minimum objective value increase.
  3. *Route improvement by resequencing*: Remove in each vehicle's route an order with maximum positive tardiness, if any, and reinsert it in a best possible way in terms of a maximum objective value decrease until no further improvement is possible.
  4. *Route improvement by reassignment*: Remove an order with maximum positive tardiness in each vehicle's route, if any, and reinsert it in another vehicle's route in a best possible way in terms of a maximum total objective value decrease until no further improvement is possible. In order to choose the vehicle which receives the order, check all vehicles and select one that leads to smallest total objective value.
  5. *Route improvement by resequencing*: Repeat step 3.
- 2OPT (2OPT, [Croes \(1958\)](#)): Obtain initial routes for each available vehicle by applying SRH. Apply to each vehicle's route the well-known 2OPT algorithm.
- SIMULATEDANNEALING (SA, [Kirkpatrick et al. \(1983\)](#)): A swap move in a route consists of exchanging the positions of two locations if a feasible sequence of pickup and delivery locations is obtained; a shift move in a route consists of shifting a number of successive locations to another position in the route if a feasible sequence of pickup and delivery locations is obtained. Obtain initial routes for each available vehicle by applying SRH. Apply to each vehicle's route the well-known SIMULATEDANNEALING scheme.
- TABUSEARCH (TS, [Cordeau and Laporte \(2003\)](#)): Let the neighborhood of a route set consist of all route sets which emanate from removing the pickup and delivery locations of an order from a first route and inserting them at best possible points of a second route in terms of a minimum objective value increase. In TS, the auxiliary objective for route quality is modified by adding a penalty proportional to the number of times that the move resulting in the neighboring route set has been applied previously. A route is tabu if it results from reinserting an order which has been removed from it no longer than a prescribed maximum number of iterations ago. Obtain initial routes for each available vehicle by applying SRH and set the current route set to this solution. Repeatedly set the current route set to a route set with minimum total objective value among all route sets in the neighborhood of the current route set such that each of the routes is non-tabu until no further improvement is made over a prescribed number of iterations.

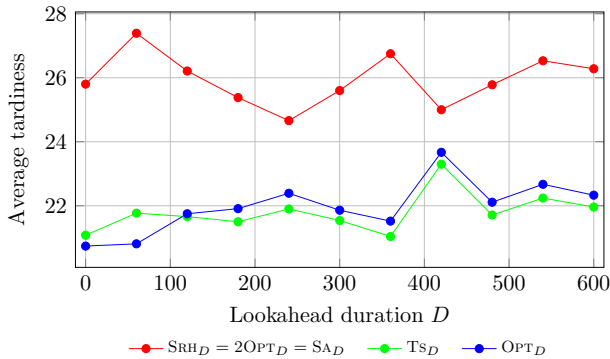


**Fig. 10** Average total distances for different lookahead durations and  $n = 50$  in the pickup and delivery service

- **OPTIMAL (OPT, Toth and Vigo (2002))**: Solve an MIP formulation of the snapshot pickup and delivery problem. Assign orders to vehicles and route them as suggested by the solution.

**Average results** We restrict attention to three selected performance criteria which already illustrate the trade-offs between competing goals: Total travel distance in kilometers, average tardiness of pickup and delivery operations in minutes, and order throughput in orders per hour. Figure 10 shows the average total distance covered by all vehicles for different lookahead durations. Apart from OPT, all algorithms acquire reductions through additional lookahead. However, because of time windows, improvement appears neither as drastic nor as reliable as in previous settings that were based on the TSP with allowed immediate service of requests (cf. Sect. 4.1 and Dunke (2014)). We attribute the major degree of unpredictability concerning the travel distance to the algorithms' rationale which also attempts to minimize average and maximum tardiness by means of the auxiliary objective function resulting from the linear combination of the three goals as discussed above. Moreover, the total distance deterioration of OPT for increasing lookahead is provoked by exact solutions of snapshot problems. Unfortunately, these exact snapshot solutions are unrobust with respect to the incorporation of new requests and may lead to additional detours.

Algorithms SRH, 2OPT and SA are found to fare best for the total distance criterion; they exhibit identical behavior which means that edge exchanging moves as well as swap and shift moves on the route set determined by SRH are ineffective. Hence, 2OPT and SA which focus on intra-route improvement (Kallrath 2005) offer no additional benefit. This is explained by the already elaborate route construction of SRH based on a best insertion policy and the difficulty of preserving feasibility upon route modifications due to time windows and precedence restrictions of pickups and deliveries. Although TS also begins with routes initially determined by SRH, the possibility of order reassignments from one route to another leads to structurally different routes that are obviously worse for the total distance criterion. Yet, for other objective functions we see that inter-route improvements (Kallrath 2005) of TS are profitable. Concerning exact reoptimization by OPT, lookahead leads to unstable routings as figured out for the TSP (Dunke 2014): Partial solutions that are advantageous for a snapshot situation



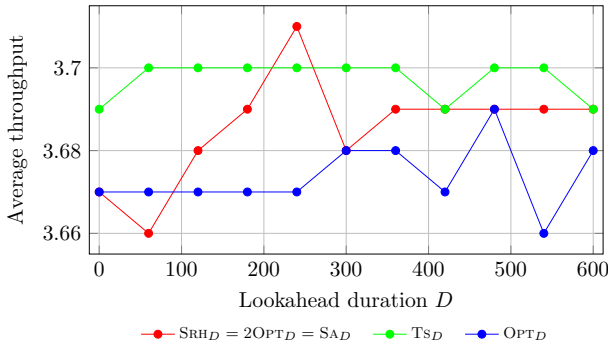
**Fig. 11** Average tardinesses for different lookahead durations and  $n = 50$  in the pickup and delivery service

may turn out disastrous when the situation changes as new transportation orders pop up.

The picture changes drastically when the average tardiness over all pickups and deliveries is considered: Fig. 11 suggests algorithms TS and OPT as the most promising candidates to keep tardiness low which is in sharp contrast to the superiority of the SRH-based algorithms with respect to the total distance criterion. We recall that route quality is assessed by an auxiliary objective which linearly combines total distance, mean tardiness and maximum tardiness. Hence, algorithms shift their focus on whatever optimization goal can be addressed best by their rationale. In this sense, TS fares best on tardiness-related objectives by trading an increase in the total distance for a decrease in the average (and maximum) tardiness.

No algorithm is found to benefit from lookahead with respect to tardiness-related goals. Quite to the contrary, even sophisticated algorithms like TS and OPT struggle with the instability of “locally” good solutions whose advantages are likely to be relinquished in the remainder of the request sequence. Instead, ad-hoc planning without too much future information is advisable because deviations between previously calculated routes and actual travel routes are likely to occur anyway. Since even in the pure online setting there are always enough unfulfilled orders to induce high vehicle utilization, it suffices to consider the transportation orders known in this case. The approximately parallel behavior of TS and OPT is due to our stipulation to use TS as a substitute for OPT in case of more than ten orders so as to guarantee reasonable computational effort. By reinspecting the average distance in Fig. 10, we also recognize the approximately parallel behavior of TS and OPT for  $D \geq 180$ . Hence, we draw the same conclusion as in order picking: When the lookahead duration exceeds 3h, subproblems become too large to be solved by OPT within 120s.

From the first two objectives, we come to the interim conclusion that the dilemma of multicriteria optimization is preserved even under large lookahead and that it is not trivial to design algorithms compliant with all objectives in a system of conflicting goals. Since creating short distance routes may only come along with large violations in the time window constraints, the decision maker has to be aware of his own trade-off relations for different goals in order to reach a final decision on algorithm quality.



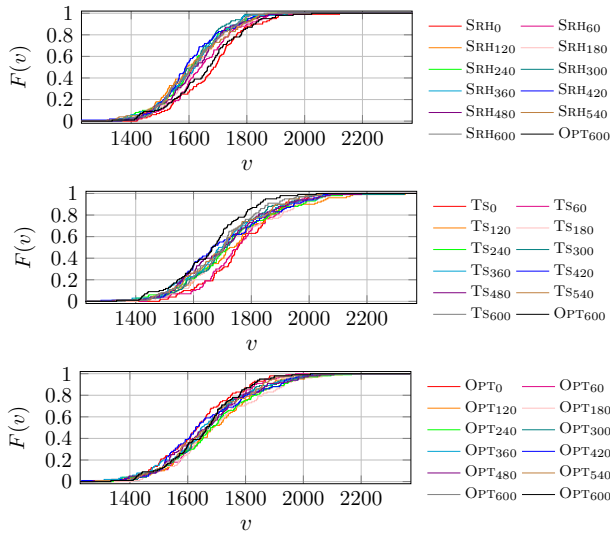
**Fig. 12** Average throughput for different lookahead durations and  $n = 50$  in the pickup and delivery service

The average throughput of transportation orders as achieved by the different algorithms is shown in Fig. 12. Looking at the scale of the diagram, we find that differences between the algorithms are only of minor magnitude. This observation is explained by the hard restriction on the earliest possible start time of pickup and delivery operations at the lower bound of corresponding time window intervals that any algorithm has to respect.

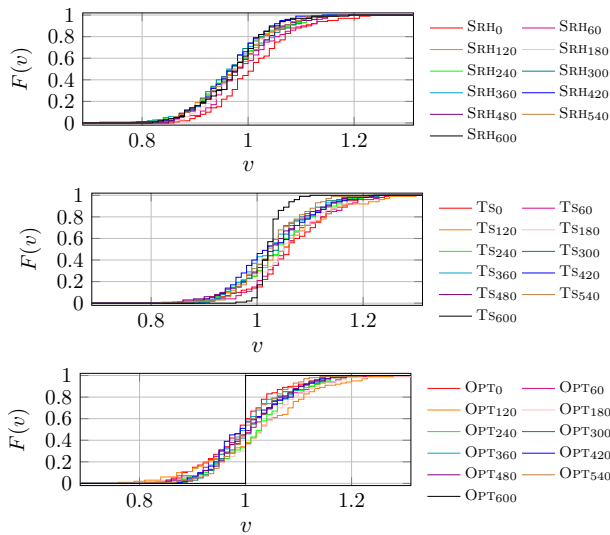
We come to the overall conclusion that in the problem setting under consideration with time windows and multiple types of unpredictable events, there is no essential benefit from lookahead in terms of major improvements in the overall route plan. Immediate planning upon arrival of transportation orders that does not account for too much future information proves to be a sufficient methodology to determine feasible routes of fair quality. The decision maker is left over with the task of choosing the algorithm candidate most consonant with his individual preferences: If goals are equally important, OPT leads to balanced results concerning several objectives; if travel distance (tardiness and maximum tardiness) minimization is considered the most important goal, SRH and OPT (TS and OPT) are the most promising candidates.

**Distributional Results.** In Figure 13, the empirical counting distribution functions of the total distance lie close to each other for different lookahead durations and there are countless intersections of the plots contradicting an exclusively positive benefit from lookahead. Instances with deteriorated total distance value are encountered every now and then, even if more lookahead was given. Nevertheless, the slight positive influence of lookahead can clearly be seen by the relative position of the plots of successive lookahead levels.

Performance ratios of the total distance incurred by the online algorithms under lookahead relative to  $TS, OPT_{600}$  in Fig. 14 appear centered around the value of 1. This means that—albeit their informational state is worse—online algorithms under lookahead lead to shorter routes on a considerable proportion of instances as compared to the routes determined under complete information. The plots of the empirical counting distribution functions exhibit numerous intersection points with each other, yet allow to establish an approximate order by their relative positions to each other for differ-



**Fig. 13** Empirical counting distribution functions of distance for  $n = 50$  in the pickup and delivery service

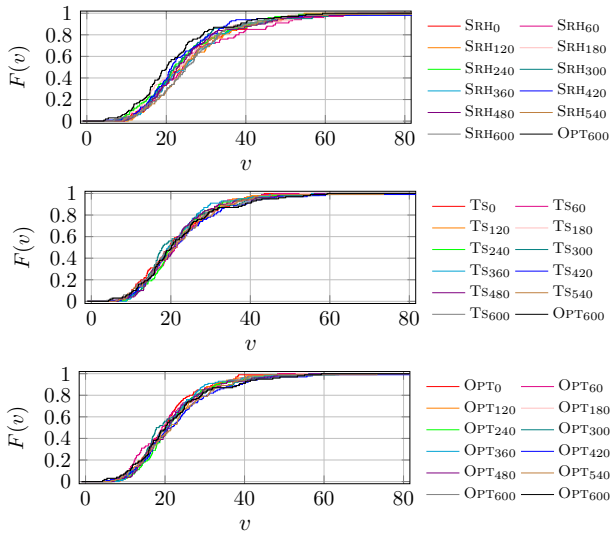


**Fig. 14** Empirical counting distribution functions of performance ratio of distance relative to TS.OPT<sub>600</sub> for  $n = 50$  in the pickup and delivery service

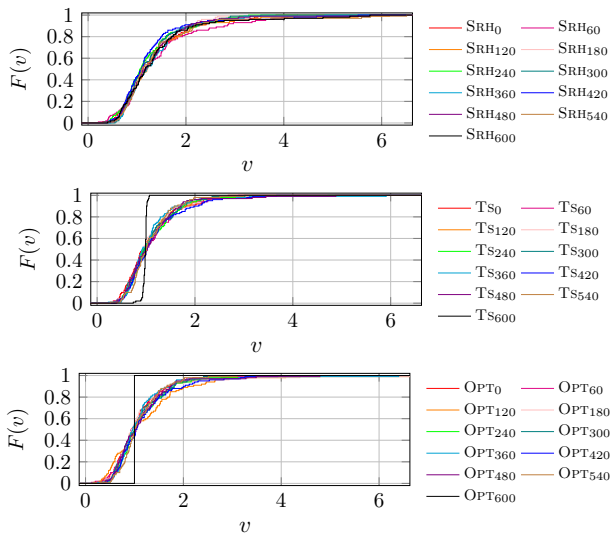
ent lookahead durations. A significant fraction of input instances has experimental competitive ratio smaller than 1.

Concerning the distributional results with respect to the tardiness over all pickups and deliveries, Figures 15 and 16 are affirmative to the ineffectiveness of additional lookahead time: Plots of all different lookahead durations intersect with each other in a





**Fig. 15** Empirical counting distribution functions of tardiness for  $n = 50$  in the pickup and delivery service



**Fig. 16** Empirical counting distribution functions of performance ratio of tardiness relative to TS.OPT<sub>600</sub> for  $n = 50$  in the pickup and delivery service

disordered fashion many times such that no order relation between any of the empirical counting distribution functions from different information regimes is recognizable.

As a result from the distributional analysis we find that the non-dominance of high lookahead durations over low lookahead durations is also found on single instances, i.e., we have an indefinite behavior of algorithms with respect to the lookahead sup-

plied. Moreover, we see that total distance values typically range between 1400 and 2200 km. Objective values throughout this range occur with significant frequency such that there is no characteristic total distance obtained by any algorithm. Concerning the observed tardinesses, we have a similar picture of variability.

We conclude this section by pointing out that hard constraints and competing objectives make it considerably harder for algorithms to elicit any improvement out of additional information. In particular, the straight-forward extensions of pure online algorithms to the lookahead case are of no use. From a managerial point of view, we recommend not to blindly install technical devices for the retrieval of lookahead information and not to hope for improvement without having an algorithm which reliably exploits lookahead in the face of competing objectives. Instead, it is advisable to spend research effort on the development of algorithms that align their rationale with lookahead and multiple goals.

### 4.3 Lessons learned

The simulation studies revealed that the first step towards efficient logistics operations consists of selecting the right algorithm which matches the decision maker's preferences best. Simulation is a suitable method to elicit the information (e.g., key performance indicators or counting distributions of algorithm performance for multiple goals) needed by managers of real world systems to successfully deploy their applications. A one-to-one transfer of statements about the lookahead effect from elementary problems (cf. also [Dunke \(2014\)](#)) to complex dynamic settings is not possible. Nevertheless, due to being embedded into the logic of the dynamic system, the elementary problems (e.g., the TSP) still deliver explanations for effects in realistic settings. For instance, the large difference in the impact of lookahead in the two applications is mainly due to the permission right for immediate processing that was invoked in the order picking system but not in the pickup and delivery service. In general, practical features may counteract the effects from the elementary problems. As a result, to a large extent a problem's constraint set already implies in advance the optimization potential exploitable by algorithms. Algorithms based on exact reoptimization showed no additional benefit compared to heuristic methods. In particular, reasonable computing times are almost surely exceeded for large lookahead durations due to the size of MIP models.

## 5 Conclusion

This paper focused on the relation between dynamic systems, especially discrete event systems, and algorithmic solution methods for decision making tasks embedded in these systems. Since, especially in the design of real world applications, comprehensive performance measurement of control strategies is needed, the first part was concerned with the methodological basis of optimization within dynamic systems. Due to the analogies between online optimization and discrete event systems, we recognized that it is possible to model online decision making as algorithmic subroutines of discrete event systems. However, because of the enormous number of dependen-

cies, we learned that realistic systems require an analysis by means of discrete event simulation instead of exact methods. Hence, simulation models represent a suitable framework for analyzing quantities of interest, in particular in the form of average and distributional analysis methods. Moreover, the usage of discrete event simulation models is now justified methodologically. We finally made practical use of the methods derived for the simulation-based analysis of online algorithms in real world systems: The compelling question of how much can be achieved through lookahead in terms of solution quality was answered in two exemplary applications using discrete event simulation and subsequent adoption of the performance measurement approaches. We recommend future research on a systematic and standardized way to include optimization into simulation environments; generic approaches have a high potential to help the optimization and simulation community to speak the same language as well as to gain a deeper and more profound general knowledge of why systems behave the way they do under given decision routines. Further, we found that real world applications typically feature multiple, oftentimes conflicting goals. Clearly, the task of designing algorithms which are able to operate in favor of more than one objective at the same time is not trivial and requires more sophisticated approaches.

## References

- Angelopoulos S, Dorrigiv R, López-Ortiz A (2007) On the separation and equivalence of paging strategies. In: Proceedings of the 18th annual ACM-SIAM symposium on discrete algorithms, pp 229–237
- Becchetti L, Leonardi S, Marchetti-Spaccamela A, Schäfer G, Vredeveld T (2006) Average-case and smoothed competitive analysis of the multilevel feedback algorithm. *Math Oper Res* 31(1):85–108
- Ben-David S, Borodin A (1994) A new measure for the study of on-line algorithms. *Algorithmica* 11(1):73–91
- Blom M, Krumke S, de Paep W, Stougie L (2000) The online TSP against fair adversaries. In: Bongiovanni G, Petreschi R, Gambosi G (eds) *Algorithms and complexity*. Springer, Berlin, pp 137–149
- Borodin A, El-Yaniv R (1998) *Online computation and competitive analysis*. Cambridge University Press, Cambridge
- Boyar J, Favrholdt L (2007) The relative worst order ratio for online algorithms. In: *ACM transactions on algorithms*, 3(2), article no. 22
- Boyar J, Favrholdt L, Larsen K, Nielsen M (2003) Extending the accommodating function. *Acta Inform* 40(1):3–35
- Boyar J, Larsen K, Nielsen M (2002) The accommodating function: a generalization of the competitive ratio. *SIAM J Comput* 31(1):233–258
- Cassandras C, Lafortune S (2008) *Introduction to discrete event systems*, 2nd edn. Springer, Berlin
- Coffman E, So K, Hofri M, Yao A (1980) A stochastic model of bin-packing. *Inf Control* 44(2):105–115
- Cordeau J, Laporte G (2003) A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transp Res B Methodol* 37(6):579–594
- Croes GA (1958) A method for solving traveling-salesman problems. *Oper Res* 6(6):791–812
- Csirik J, Woeginger G (2002) Resource augmentation for online bounded space bin packing. *J Algorithms* 44(2):308–320
- Dorrigiv R, Lopez-Ortiz A (2007) Adaptive analysis of on-line algorithms. In: Fekete S, Fleischer R, Klein R, Lopez-Ortiz A (eds) *Robot navigation*, number 06421 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI). Schloss Dagstuhl, Germany
- Dorrigiv R, López-Ortiz A (2008) Closing the gap between theory and practice: new measures for on-line algorithm analysis. In: Proceedings of the 2nd international conference on algorithms and computation, pp 13–24
- Dorrigiv R, López-Ortiz A, Munro J (2009) On the relative dominance of paging algorithms. *Theor Comput Sci* 410(38–40):3694–3701
- Dunke F (2014) *Online Optimization with Lookahead*. Ph.D. thesis, Karlsruhe Institute of Technology

- Fiat A, Woeginger G (1998) Competitive odds and ends. In: Fiat A, Woeginger G (eds) *Online algorithms: the state of the art*. Springer, Berlin, pp 385–394
- Franaszek P, Wagner T (1974) Some distribution-free aspects of paging algorithm performance. *J ACM* 21(1):31–39
- Ghiani G, Laporte G, Musmanno R (2004) *Introduction to logistics systems planning and control*. Wiley, Hoboken
- Grötschel M, Krumke S, Rambau J (eds) (2001) *Online optimization of large scale systems*, Springer
- Grötschel M, Krumke S, Rambau J, Winter T, Zimmermann U (2001) Combinatorial online optimization in real time. In: Grötschel M, Krumke S, Rambau J (eds) *Online optimization of large scale systems*. Springer, Berlin, pp 679–704
- Henn S, Koch S, Wäscher G (2012) Order batching in order picking warehouses: a survey of solution approaches. In: Manzini R (ed) *Warehousing in the global supply chain*. Springer, Berlin, pp 105–137
- Hiller B (2009) *Online optimization: probabilistic analysis and algorithm engineering*. Ph.D. thesis, Technische Universität Berlin
- Huber C (2011) *Throughput analysis of manual order picking systems with congestion consideration*. Ph.D. thesis, Karlsruher Institut für Technologie
- Jaynes E (1957) Information theory and statistical mechanics. *Phys Rev* 106(4):620–630
- Jaynes E (1957) Information theory and statistical mechanics II. *Phys Rev* 108(2):171–190
- Kallrath J (2005) *Online storage systems and transportation problems with applications: optimization models and mathematical solutions*. Springer, Berlin
- Kalyanasundaram B, Pruhs K (2000) Speed is as powerful as clairvoyance. *J ACM* 47(4):617–643
- Karlin A, Manasse M, Rudolph L, Sleator D (1988) Competitive snoopy caching. *Algorithmica* 3(1–4):79–119
- Karlin A, Phillips S, Raghavan P (2000) Markov paging. *SIAM J Comput* 30(3):906–922
- Kenyon C (1996) Best-fit bin-packing with random order. In: *Proceedings of the 7th annual ACM-SIAM symposium on discrete algorithms*, pp 359–364
- Kirkpatrick S, Gelatt CD, Vecchi P (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
- Koutsoupias E, Papadimitriou C (2000) Beyond competitive analysis. *SIAM J Comput* 30(1):300–317
- Krumke S, Laura L, Lipmann M, Marchetti-Spaccamela A, Paepé Wd, Poensgen D, Stougie L (2002) Non-abusiveness helps: An  $o(1)$ -competitive algorithm for minimizing the maximum flow time in the online traveling salesman problem. In: *Proceedings of the 5th international workshop on approximation algorithms for combinatorial optimization, APPROX '02*, 200–214, Springer
- Lawler E, Lenstra J, Rinnooy Kan A, Shmoys D (eds.) (1985) *The traveling salesman problem: a guided tour of combinatorial optimization*. Wiley
- Miller C, Tucker A, Zemlin R (1960) Integer programming formulation of traveling salesman problems. *J ACM* 7(4):326–329
- Müller A, Stoyan D (2002) *Comparison methods for stochastic models and risks*. Wiley, Hoboken
- März L, Krug W (2011) Kopplung von Simulation und Optimierung. In: Krug W, Rose O, Weigert G (eds) *Simulation und Optimierung in Produktion und Logistik: Praxisorientierter Leitfaden mit Fallbeispielen*. Springer, Berlin, pp 41–45
- Psaraftis H (1995) Dynamic vehicle routing: status and prospects. *Ann Oper Res* 61(1):143–164
- Raghavan P (1991) A statistical adversary for on-line algorithms. *DIMACS Ser Discrete Math Theor Comput Sci* 7:79–83
- Scharbrodt M, Schickinger T, Steger A (2006) A new average case analysis for completion time scheduling. *J ACM* 53(1):121–146
- Sleator D, Tarjan R (1985) Amortized efficiency of list update and paging rules. *Commun ACM* 28(2):202–208
- Spielman D, Teng S (2004) Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. *J ACM* 51(3):385–463
- Stadler H, Kilger C (eds.) (2008) *Supply chain management and advanced planning: concepts, models, software, and case studies*. Springer, 4th edn
- Toth PM, Vigo D (eds.) (2002) *The vehicle routing problem*. SIAM
- Verein Deutscher Ingenieure (VDI) (1996) VDI-Richtlinie 3633. Simulation von Logistik-, Materialfluß- und Produktionssystemen: Begriffsdefinitionen. In: *VDI-Handbuch Materialfluß und Fördertechnik*, Beuth
- Young N (1994) The  $k$ -server dual and loose competitiveness for paging. *Algorithmica* 11(6):525–541