Simulation in Produktion und Logistik
Entscheidungsunterstützung von der Planung bis zur Steuerung
Wilhelm Dangelmaier, Christoph Laroque & Alexander Klaas (Hrsg.)
Paderborn, HNI-Verlagsschriftenreihe 2013

Simulative Algorithm Analysis in Online Optimization with Lookahead

Simulative Algorithmenanalyse für Online-Optimierungsprobleme mit Lookahead

Fabian Dunke, Stefan Nickel, Karlsruhe Institute of Technology, Karlsruhe (Germany), fabian.dunke@kit.edu, stefan.nickel@kit.edu

Abstract: In online optimization, input data is revealed sequentially. Optimization problems from practice exhibit this type of information disclosure as opposed to standard offline optimization where all data is known in advance. We present a general method to analyse the impact of lookahead on the solution quality obtained by algorithms in applications of online optimization with lookahead. To this end, we first propose a generic framework to harmonize concepts and notation throughout domains. In addition, we tailor performance measurement approaches to the needs of online algorithms under various lookahead levels. In complex settings, the method is best applied within a sample-based simulation scheme to judge on algorithm quality and to determine the value of additional information. Finally, we instantiate the framework for three applications and conduct numerical experiments. Results indicate that algorithm behaviour depends on the amount of lookahead, but also that the lookahead value strongly relies on the problem itself.

1 Introduction

Due to technological advances, the number of problem settings where input data is obtained and processed in real-time is continuously increasing. Examples include storage and retrieval, order picking, machine scheduling, or distribution planning (Jaillet and Wagner 2006; Albers 2003). Since these systems are subject to steady information disclosure, they are said to be online. Optimization problems arising in these contexts are called online optimization problems and algorithms for them have to operate dynamically. This paradigm is completely opposite to that of classical (offline) optimization where all input data is assumed to be known in advance. In between these two extremes, there is an intermediate setting which we will call online optimization with lookahead. Here, the amount of accessible information is governed by a lookahead mechanism, e.g., preponing bar code scanners or sending order information at an earlier time.

The function logic in operating and controlling dynamic systems repeatedly requires decision making in order to continue. For each of these decisions, an online algorithm is called as a subroutine. It has to determine partial solutions based on the currently available input data such that the overall solution which will be composed of all partial solutions will be as good as possible. Figure 1 sums up the hierarchical relation between the logics in a dynamic system and the online optimization module needed therein (März and Krug 2011; Lavrov and Nickel 2005).

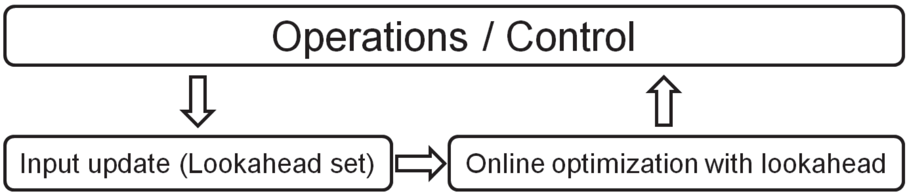

Figure 1: *Hierarchical relation between the function logic of operating / controlling a dynamic system and the module of online optimization with lookahead*

Clearly, an exact analysis of dynamic systems requiring repeated decision making is possible only for easy and small problem settings. For more complex settings, we use simulation as a method to duplicate the function logic of the real-world system in an abstract way (Cassandras and Lafortune 2008; VDI 1993).

Both in designing and operating a dynamic system, preferably a best possible algorithm for each decision needs to be identified within a set of viable algorithm candidates. As the system evolves over time and decisions are based on the currently available amount of information, we also ask for an algorithm's sensitivity to information with respect to the outcome. The following two central questions will serve as a guideline subsequently:

- Which one of the algorithm candidates for a given problem is most suitable?
- How is an algorithm's performance affected by the amount of lookahead?

In order to answer these questions on a simulative basis, we may run a sufficiently large number of independent simulations replications to obtain empirical results. We call this type of analysis simulative algorithm analysis.

2 A Framework for Online Optimization with Lookahead

We consider how a given system whose function logic iteratively requires solving an online optimization problem can be tackled formally and domain-independent. To this end, let the input of an optimization problem be given in form of an input sequence $\sigma = (\sigma_1, \sigma_2, ..., \sigma_n)$. The elements of σ are called input elements, and the set of unprocessed known input elements at time t is called the lookahead set at time t.

2.1 Lookahead Types and Processing Characteristics

Which elements of σ are known exactly to an algorithm depends on the lookahead type; which elements of σ are eligible to be processed next depends on the processing characteristics. Lookahead type and processing characteristics are intertwined by the (physical) restrictions imposed by the dynamic system.

2.1.1 Lookahead Types

Numerous lookahead concepts have been introduced with reference to a specific problem setting (Allulli et al. 2008; Jaillet and Wagner 2006; Albers 1997; Breslauer 1996). The following two definitions seem intuitive and are not specific to a problem. They cover a multitude of the concepts proposed in literature.

- In request lookahead, at each point in time a fixed number K of future input elements is seen. Examples can be found in paging systems of computer programs, in material provision in production lines or in pallet embarkation.
- In time lookahead, at each point in time t the algorithm knows those input elements which would have been known by time $t + D$ if there was no lookahead. D is called lookahead duration. Examples for this type of lookahead can be found in vehicle routing, in pick list generation of order picking systems and in dynamic passenger information boards.

The classification into request lookahead and time lookahead is not exhaustive, but these two types represent the most important prototypes for the definition of further lookahead types, e.g., time lookahead with maximum number of input elements.

2.1.2 Processing Characteristics

Besides the mode of information disclosure itself, we have to take into account the implications of the input release mechanism on the processing of the input elements.

Processing order:

- In random access processing order, all known input elements from the lookahead set are eligible to be processed in arbitrary order. An algorithm does not only profit from the preponed release of an input element but also from the right to permute their processing order as compared to the case without lookahead.
- In sequential processing order, all known input elements from the lookahead set have to be processed in their order of release. An algorithm only profits from information being available earlier, but processing order changes are forbidden.

Processing availability:

- In immediate processing availability, an input element is ready to be (physically) processed directly upon receiving the respective input element information.
- In regular processing availability, it has to be waited until the regular earliest processing time of the pure online setting has been reached before processing of the input element is started.
- In intermediate processing availability, processing an input element is possible at some time between its notification and its regular earliest processing time.

Note that the processing characteristics determine some of the constraints under which an online algorithm with lookahead has to operate. Hence, their influence on the outcome of an algorithm for a given problem instance can be enormous.

2.2 Process Model of Online Optimization with Lookahead

Interpreting the arrival of a new input element σ_i as an event, we describe the system as a discrete event system, and take into account that state transitions are also triggered by decisions of an online optimization algorithm immediately following the notification of a new input element.

2.2.1 Modelling Elements

We embed the system into a time component and denote current time by t. Each input element has a (reference) release time in absence of lookahead, and in presence of lookahead this release time is preponed as governed by a lookahead device. The process model comprises the following elements:

- The input sequence $\sigma = (\sigma_1, \sigma_2, ...)$ holds the successively arriving input elements.
- The lookahead set L_t contains all input elements that have been released up to time t which are yet unprocessed or still in processing.
- The state space S comprises the set of configurations that can be reached by the system including the current lookahead set. Note that the state may change continuously. In order to inspect the system only at discrete points in time, we introduce objective states.
- The objective state space O extracts all information regarding the development of the objective value. We assume that the objective state can only change as a result of finished processing of an input element at discrete time instants.
- The action space A is the union of the set of all actions which can be performed on an input element with the null-action which is assigned to an input element as long as no action has been determined for it. An action describes the processing that an input element is about to experience.
- The objective state transition function f: $S \times O \rightarrow O$ determines the successor objective state of a given current objective state $o \in O$ which results by finished processing of an input element's action.
- An algorithm ALG is a goal-oriented sequence of computations to determine the (intended) action for each input element in the lookahead set.

2.2.2 Process Model

The process model for online optimization problems with lookahead describes the temporal interaction of the modeling elements. The evolution of the system state and objective state trajectory is driven by input element releases and by performing actions on the input elements as determined by algorithm ALG. The process model implements the function logic of a dynamic system by iteratively performing the steps in the flow chart of Figure 2:

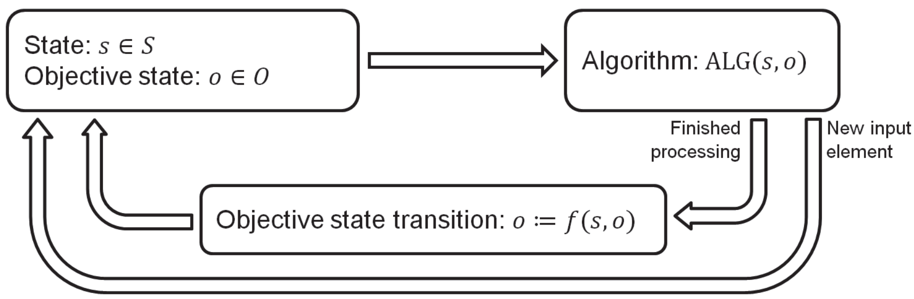

Figure 2: *Process model for online optimization with lookahead*

2.3 Framework for Online Optimization with Lookahead

The generic framework for online optimization with lookahead comprises the combination of lookahead type, implied processing characteristics and process model. Its instantiations give a clear description of the constraints and the potential workflow that the system under consideration undergoes while processing an arbitrary input sequence using an algorithm with lookahead capabilities.

The framework harmonizes the representations of the solution process in online optimization problems with lookahead and unifies notation throughout application domains. It facilitates systematic algorithm analyses in arbitrary applications using the same paradigms concerning the solution process and the lookahead mechanisms.

Using a generic implementation of the process model and concrete instantiations of lookahead type, processing characteristics and algorithms, we analyzed the lookahead effect for algorithms in paging, bin packing, scheduling and routing problems (cf. also section 4).

3 Algorithm Analysis in Online Optimization with Lookahead

Due to the unknown future, no algorithm can solve an online optimization problem to optimality on any instance. Hence, in order to compare the quality of algorithms, we have to agree upon a performance measurement method which is capable of conveying the algorithms' qualities and assessing the value of additional lookahead. Moreover, we wish to apply the method within a simulation environment.

3.1 Survey of Performance Measures in Online Optimization

The standard performance measure in online optimization is the competitive ratio (Borodin and El-Yaniv 2005; Woeginger and Fiat 1998). An online algorithm in a minimization problem is c-competitive if for all σ it holds that $\mathrm{ALG}(\sigma) \leq c \cdot \mathrm{OPT}(\sigma)$ where $\mathrm{OPT}(\sigma)$ is the minimal cost of an optimal offline algorithm for processing σ.

The competitive ratio yields a worst case analysis where a pathological input sequence may decide upon its value. In many cases, this type of analysis leads to overly pessimistic results (Hiller and Vredeveld 2012; Boyar et al. 2009) and it does not give a clue at an algorithm's average and overall performance since it only rates an algorithm based on a single number. In practice, choosing this performance measure can lead to rude surprises concerning the behaviour of an algorithm on typical input instances, and bad algorithms may be favoured. Also, this type of analysis is impossible in slightly more complex systems (Woeginger and Fiat 1998).

There are no special approaches for the case with lookahead. Existing research is concentrating on how the competitive ratio improves as a result of lookahead – if that (Allulli et al. 2008; Jaillet and Wagner 2006; Albers 1997; Breslauer 1996).

3.2 Holistic Performance Measurement

We present two distributional methods which – taken together – are well-suited for comprehensively analysing the performance of online algorithms under lookahead within a simulation environment. We subsume all candidate algorithms underlying the same lookahead mechanism in one algorithm set.

3.2.1 Distribution of Performance Ratio for Algorithm Sets

Let A and B be sets of algorithms, let σ be an input sequence and let ALG(σ) denote the objective value of algorithm ALG on input sequence σ, then the ratio

$$c_\sigma^{B/A} := \frac{\min_{\mathrm{ALG}' \in B} \mathrm{ALG}'(\sigma)}{\min_{\mathrm{ALG} \in A} \mathrm{ALG}(\sigma)} \tag{1}$$

is called performance ratio of B relative to A with respect to σ.

We can identify A as a set of algorithms without lookahead and B as a set of algorithms with lookahead. For a sufficiently large number of input sequences, we can derive the (empirical) performance ratio distribution to get an impression over a broad spectrum of instances (Fig. 3a)). The performance ratio is a local measure in the sense that it compares algorithm sets directly on the same input sequence.

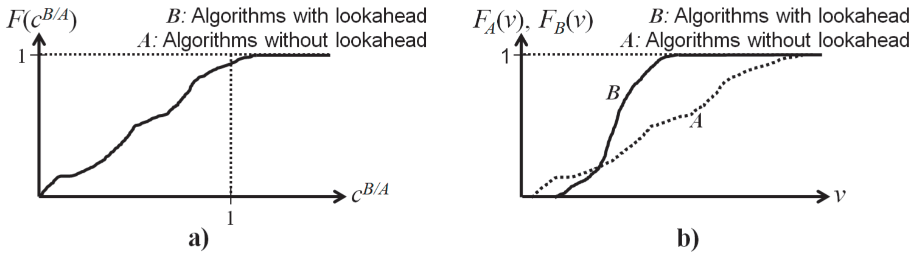

Figure 3: *Distribution functions F(·) of a) performance ratio and b) objective value*

3.2.2 Distribution of Objective Value for Algorithm Sets

Let A be a set of algorithms and let σ be an input sequence, then the value

$$v_\sigma^A := \min_{\mathrm{ALG} \in A} \mathrm{ALG}(\sigma) \tag{2}$$

is called the objective value of algorithm set A with respect to σ. For a sufficiently large number of input sequences, we can derive (empirical) distributions of the objective value to get an impression over a broad spectrum of instances (Fig. 3 b)). The objective value distribution is a global performance measure for algorithm set A and algorithm set B, respectively, in the sense that it portrays the individual behaviour of A and B over the input sequences. A comparison between two algorithm sets is made indirectly by comparing the shape and position of the plots.

4 Numerical Results

We instantiated the generic framework in three applications to perform numerical experiments using the performance measures from above. In the subsequent simulative algorithm analyses, we focus on finding an appropriate lookahead level. Simulation models were implemented in *C++* (first / third application) and *AnyLogic 6* (second application), integer programming (IP) problems were solved by *CPLEX 12.4*.

4.1 Online Traveling Salesman

The traveling salesman problem (TSP) is the core of numerous transportation and routing problems. It is defined by n points in a metric space that have to be visited by a server tour such that the tour length is as small as possible. Both request and time lookahead can be applied.

Algorithms: We investigate algorithms under request lookahead of size K.

- NEAREST NEIGHBOR: The closest point in the lookahead set is visited next.
- INSERTION: The next point visited by the server is the first point of a Hamilton path which is formed by successive best insertion of the points in the lookahead.
- 2-OPT / 3-OPT: The next point visited by the server is the first point of a Hamilton path which is formed by successively improving an initial tour through exchanges of edge pairs (2-OPT) and edge triples (3-OPT), respectively.
- SIMULATED ANNEALING: Identical to 2-OPT except that temporarily worse tours are allowed to move from local to global optima of the snapshot problem.
- IP SOLVE: The next point visited by the server is the first point of a Hamilton path which is determined by solving an IP formulation of the snapshot problem.

Average results: For 25 and 100 points, we ran 100 simulation replications for each algorithm and lookahead. The average tour length for different lookahead sizes demonstrates the huge benefit accrued by lookahead (Fig. 4). For K=5 and 100 points, tour length reduces to 50% as compared to the online setting, and for K = 100 to 12%. Surprisingly, all algorithms experience positive effects in a comparable amount. The easy NEAREST NEIGHBOR algorithm induces stability in the sense that it guarantees no jumps between regions. For algorithms working towards local optimality such as IP SOLVE these jumps may occur and prove globally inefficient.

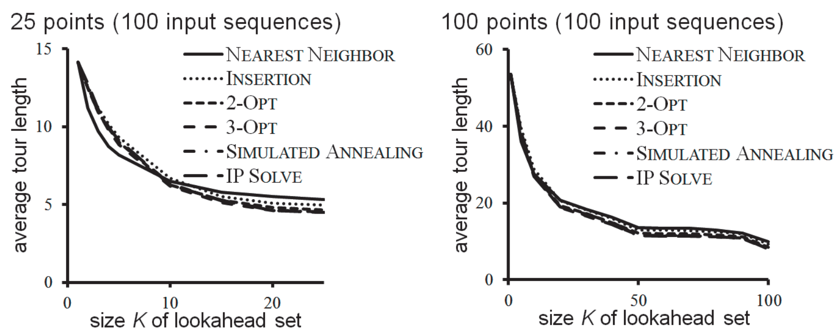

Figure 4: *Average tour length with 25 points and 100 points*

Distributional results: Figure 5 shows the distribution of the performance ratio and objective value for algorithm sets with selected lookahead regimes for 100 points relative to pure online algorithms. The plots show the expected ordering (dominance) of the distributions according to the lookahead level. All plots are very steep in a characteristic area of the abscissa which accounts for a typical tour length reduction of a lookahead level.

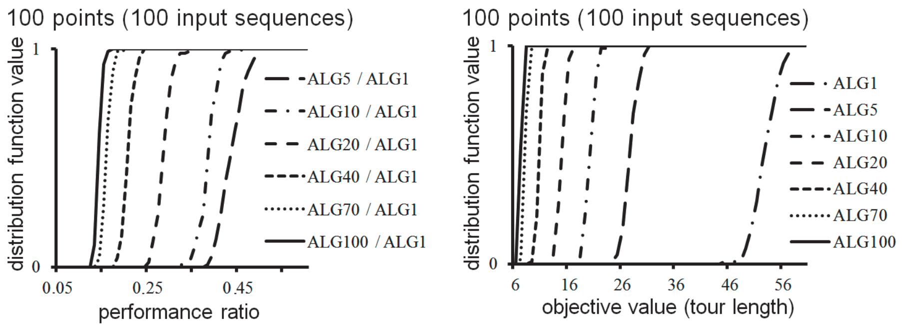

Figure 5: *Distribution functions of performance ratio and tour length*

4.2 Online Pickup and Delivery

A simulation model has been put up for a pickup and delivery service with a fleet of *N* vehicles in Karlsruhe. It covers stochastic request generation with time windows, alternative route selection, stochastic congestion information, loading and unloading times. A number of objectives (distance, lateness, load and driving time violations) were evaluated under time lookahead. We discuss only total distance.

Algorithms: We investigate algorithms under time lookahead of length *D*.

- RULE-BASED HEURISTIC: Routes are determined in four steps: Assignment of orders to vehicles based on simple criteria, resequencing of routes, reassignment of critical requests, and resequencing of routes.
- 2-OPT: The routes obtained by RULE-BASED HEURISTIC are improved using exchanges of edge pairs which lead to better feasible tours.
- SIMULATED ANNEALING: Identical to 2-OPT except that temporarily worse tours are allowed to move from local to global optima of the snapshot problem.
- TABU SEARCH: Routes obtained by RULE-BASED HEURISTIC are improved by moving to the best non-tabu route plan in the neighborhood of the current plan. A neighbor of a route plan is obtained by reassigning one request to another vehicle. The current route plan is tabu for a given number of iterations.
- IP SOLVE: Each vehicle follows a route which is determined by solving the IP formulation of the snapshot problem with all unserved requests.

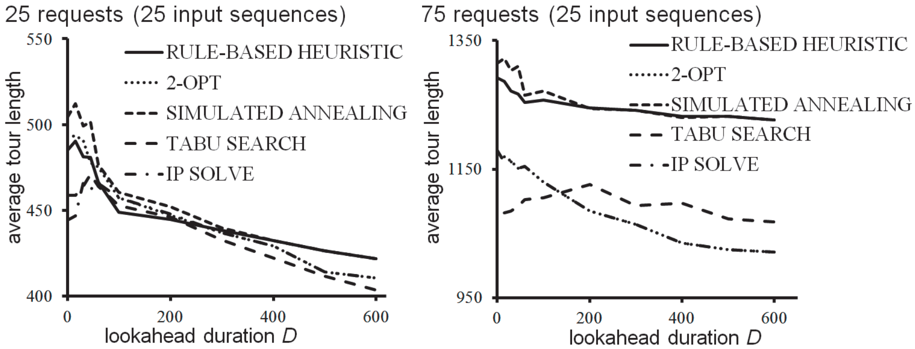

Figure 6: Average tour length with 25 requests and 75 requests

Average results: For $N = 3$ vehicles, we considered 25, 50, 75 and 100 requests per day, and lookahead between 15 and 600 minutes. For each combination, 25 simulation replications were run. Figure 6 displays that lookahead is beneficial for all algorithms albeit not as drastically as in the TSP. TABU SEARCH and IP SOLVE lead to the best routes. Small lookahead does not lead to reliable improvements as requests may pop up shortly before their pickup time window starts.

Distributional results: In Figure 7, the positions of the distribution functions confirm that lookahead is *reliably* helpful only for D large enough. Performance ratios larger than 1 indicate that an algorithm may fail to interpret lookahead to its advantage. Yet, the proportion of these instances is small. Distributions are not ordered perfectly.

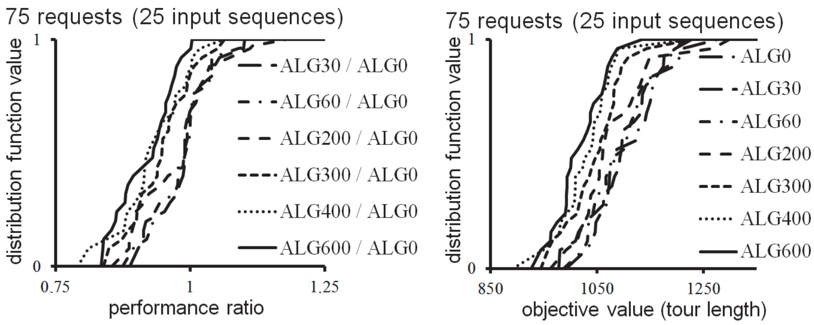

Figure 7: Distribution functions of performance ratio and tour length

4.3 Online Bin Packing

We adapt well-known online and offline algorithms (Borodin and El-Yaniv 2005) to request lookahead of size K, and also employ an exact IP approach.

Lookahead leads to fewer bins for all algorithms. However, the savings is comparatively small such that between the extreme cases of the online and offline situation only a reduction of at most 5% can be realized. Computing time intensive IP SOLVE is advantageous only when K is large enough, and there is no evidence that locally optimal solutions carry over to the global solution after all items have been packed. Hence, simple heuristics like BEST FIT or FIRST FIT are sufficient.

In the case of 1000 items, for $K \geq 500$ the distributions coincide and there is no additional value. For $K < 500$ there is an order (dominance) of the distributions confirming the (small) lookahead effect. Distributions have negligible variance.

4.4 General Conclusion

In contrast to bin packing, a strong lookahead effect is observed in the TSP, and a fair one in the pickup and delivery service. The core reason for the differences is that changing the visiting order of the points in a TSP immediately affects the total travel distance; in contrast, an item in bin packing occupies the same capacity regardless of the packing order. Moreover, we find that an algorithm's degrees of freedom as imposed by the restrictions play an important role in its ability to take advantage of lookahead. Hence, the results demonstrate that information sensitivity is strongly related to the problem itself and cannot be presupposed in general.

5 Summary and Outlook

In order to derive well-informed statements about the quality of algorithm candidates and the value of information, we presented a holistic method of algorithm analysis in online optimization with lookahead based on a common understanding of the implications of lookahead as manifested in the generic framework. The main advantage over existing methods is that an algorithm is no more compared to an almighty offline optimum but to a comparable information regime. However, the approach is mainly suitable only for empirical (approximate) analyses. Due to problem complexities exact analyses are impossible for more complex problems. The method could be further improved by incorporating an inaccuracy measure for the empirical distributions. Numerical experiments from three applications provide promising results concerning the application of the approach within a simulation environment. An interesting future research direction would be to determine exact analytical expressions for the distributions of performance ratio and objective value in simple settings, or to derive error bounds on the approximations obtained by simulation. The performance measurement approach is currently applied to other dynamic logistics systems such as order picking and machine shop scheduling.

References

Albers, S.: On the influence of lookahead in competitive paging algorithms. Algorithmica 18 (1997) 3, pp. 283-305.

Albers, S.: Online algorithms: a survey. Math. Programming 97 (2003) 1-2, pp. 3-26.

Allulli, L.; Ausiello, G.; Laura, L.: On the power of lookahead in on-line server routing problems. Theoretical Computer Science 408 (2008) 2-3, pp. 116-128.

Boyar, J.; Irani, S.; Larsen, K. S.: A comparison of performance measures for online algorithms. In: Dehne, F; Gavrilova, M.; Sack, J.-R.; Tóth, C. D. (Eds.): Algorithms and Data Structures. Berlin, Heidelberg: Springer 2009, pp. 119-130.

Breslauer, D.: On competitive on-line paging with lookahead. In: Puech, C. (Ed.): STACS 96 - Proceedings of the 13th Annual Symposium on Theoretical Aspects of Computer Science. Berlin, Heidelberg: Springer 1996, pp. 593-603.

Borodin, A.; El-Yaniv, R.: Online computation and competitive analysis. 1. pbk. ed. Cambridge, New York: Cambridge University Press 2005.

Cassandras, C. G.; Lafortune, S.: Introduction to discrete event systems. 2. ed. New York: Springer 2008.

Hiller, B.; Vredeveld, T.: Probabilistic alternatives for competitive analysis. Computer Science Research and Development 27 (2012) 3, pp. 189-196.

Jaillet, P.; Wagner, M. R.: Online routing: Value of advanced information as improved competitive ratios. Transportation Science 40 (2006) 2, pp. 200-210.

Lavrov, A.; Nickel, S.: Simulation und Optimierung zur Planung von Kommissionierungssystemen. VDI-Seminar, 2005. Ceded for use by Nickel, S.

März, L.; Krug, W.: Kopplung von Simulation und Optimierung. In: März, L.; Krug, W.; Rose, O.; Weigert, G. (Eds.): Simulation und Optimierung in Produktion und Logistik. Berlin, Heidelberg: Springer 2011, pp. 41-45.

VDI: VDI-Richtlinie 3633 Blatt 1: Simulation von Logistik-, Materialfluss- und Produktionssystemen. In: VDI (Ed.): VDI-Handbuch Materialfluss und Fördertechnik Band 8. Berlin: Beuth 1993.

Woeginger, G.; Fiat, A.: Online algorithms - State of the art. Berlin: Springer 1998.