Omega 63 (2016) 134–153

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

# A general modeling approach to online optimization with lookahead ☆

Fabian Dunke *, Stefan Nickel

*Karlsruhe Institute of Technology, Institute of Operations Research, Karlsruhe, Germany*

ARTICLE INFO

*Article history:*
Received 16 October 2014
Accepted 14 October 2015
Available online 22 October 2015

*Keywords:*
Online optimization
Lookahead
Discrete event system
Algorithm analysis

ABSTRACT

A vast number of real world problems are coined by an information release over time and the related need for repetitive decision making over time. Optimization problems arising in this context are called online since decisions have to be made although not all data is known. Due to technological advances, algorithms may also resort to a limited preview (lookahead) on future events. We first embed the paradigm of online optimization with lookahead into the theory of optimization and develop a concise understanding of lookahead. We further find that the effect of lookahead can be decomposed into an informational and a processual component. Based on analogies to discrete event systems, we then formulate a generic modeling framework for online optimization with lookahead and derive a classification scheme which facilitates a thorough categorization of different lookahead concepts. After an assessment of performance measurement approaches with relevance to practical needs, we conduct a series of computational experiments which illustrate how the general concept of lookahead applies to specific instantiations and how a knowledge pool on lookahead effects in applications can be built up using the general classification scheme.



## 1. Introduction

Although there is an agreement on the importance of coping with unexpected events in today's systems for production and logistics [26,51], recent implementations of planning and scheduling systems still suffer from their deficiency in dealing with uncertainty over time. In a rolling horizon, plans are determined on the basis of forecasts by offline optimization methods [51]. However, since only decisions of the next period are implemented before the problem gets resolved with updated data, this approach exhibits large redundancies.

On the other hand, possibilities for collection of data about near-future events are steadily increasing due to technological developments [26] such as radio frequency identification (RFID), global positioning systems (GPS) or geographical information systems (GIS). Since planning systems in these environments are subject to permanent information inflow, they are said to be online. Optimization problems in this context are called online optimization problems [24]. These problems are characterized by the fact that decisions are required to be made repeatedly before all data is available. In contrast to other methodologies for optimization under uncertainty, there are no forecasts or probabilities of future events assumed in online optimization. However, as a result of technological opportunities given above, we can now cope with uncertainty differently. Through the installation of lookahead devices, it is possible to acquire data about future events at an earlier point in time. Hence, uncertainty is tackled forcefully because parts of the previously uncertain future can now be fixed to certainty through the utilization of lookahead. Thus, the decision making process consists of repetitive decisions where the input to each decision only consists of the small, but certain part of the future known at that time. Though, as can be seen from the different information gathering devices mentioned above, it may be reasonable to be more precise with respect to the actual degree of “onlineness” in a specific problem setting. The need for a concise notion of lookahead is also reflected by the manifold perceptions of lookahead depending on the application [2–4,14,17,29,37,45,52,56]. For this reason, this paper coins the notion of online optimization with lookahead on a formal basis.

The task of solving online optimization problems is a recurring pattern in industrial applications (Fig. 1): each time the functional logic of a dynamic system requires a decision, an online algorithm is called to deliver it, i.e., partial answers based on currently available data have to be given such that the overall solution will be as good as possible.

Solution methodologies for the different optimization paradigms strongly differ from each other. Consider the input sequence

☆ This manuscript was processed by Associate Editor Ghate.

* Corresponding author.
*E-mail addresses:* fabian.dunke@kit.edu (F. Dunke), stefan.nickel@kit.edu (S. Nickel).

http://dx.doi.org/10.1016/j.omega.2015.10.009
0305-0483/

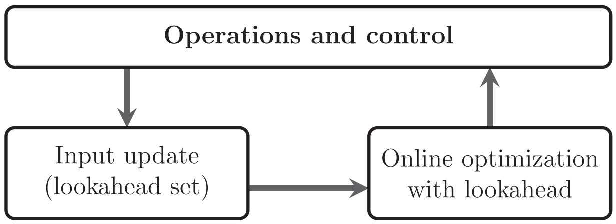


**Fig. 1.** Hierarchical relation between operations and control of a dynamic system and online optimization with lookahead.

$\sigma = (\sigma_1, \sigma_2, \ldots)$. In offline optimization, $\sigma$ is known in advance and a plan for how to process its elements can be computed directly. In the sequential model of online optimization [30], only one input element is known at a time and input elements must be processed in release order, i.e., input element $\sigma_i$ is processed based upon knowledge of $\sigma_1, \ldots, \sigma_i$ and previous decisions on $\sigma_1, \ldots, \sigma_{i-1}$. In the time-stamp model [30], each input element is assigned an arrival date such that input elements may accumulate naturally and automatically form some lookahead set of unprocessed input elements. In online optimization with request lookahead [2,52], more than one unprocessed input element may be known at a time and also an explicit formulation of processing restrictions is required. One could insist on sequential processing ($\sigma_i$ must be processed before $\sigma_{i+1}$) or allow for a processing in arbitrary order ($\sigma_{i+1}$ can be processed before $\sigma_i$). There are a plenty of variations of how lookahead is understood and what it means for the processing of single input elements. For this reason, this paper will provide the tools for classifying the main features of a specific online optimization problem with lookahead.

Literature focuses on a worst-case type analysis of algorithm performance – called competitive analysis [13,40] – where an online algorithm has to compete with an optimal offline algorithm. Derivations of this measure are based on the individual taxonomy in a specific problem and not on a general notation valid for all problems. Likewise, the case with lookahead has been addressed only rarely in specific problems from routing and transportation [4–6,36,37,52], scheduling [18,44–47,54,57,58], organization of data structures [2,3,14,41,53,55,56], data transfer [20,34], packing [29,31], lot sizing [1], metrical task systems [9,42] or graph theory [17,32,35]. To the best of our knowledge, there have been no attempts to formalize different degrees of available information in a general framework. A reason for the lack of general concepts lies in the different possibilities to deal with temporal aspects [30]. When only the order of input element releases matters, time is already modeled implicitly through the indices of $\sigma_1, \sigma_2, \ldots$ On the other hand, when time durations play a role, time aspects have to be modeled explicitly as part of the data belonging to $\sigma_1, \sigma_2, \ldots$, e.g., in the form of release times $t_1, t_2, \ldots$ This issue also accounts for various perceptions of lookahead along with its implied processing characteristics.

### 1.1. Lookahead and related concepts for uncertainty

The idea of online optimization with lookahead is based on known deterministic information previews, and hence it can be distinguished from other approaches for optimization under incomplete information or uncertainty. In stochastic programming [12], probability distributions for scenarios that take into account all uncertain factors (often in form of parameters) are known and solution quality is typically evaluated by average-case measures to immunize the solution probabilistically to incomplete information. In addition, stochastic programming is rather concerned with sporadic than with frequent decision making. Scenarios are often coined by the realization of parameter values which are considered to be some random variable. Online optimization differs from this approach strongly since it is not focused on scenarios and/or parameters, but rather on the realizations of input elements for which no stochastic principles are known to hold. Dynamic programming [8] assumes that an optimization problem exhibits the property of optimal substructures (i.e., an optimal solution is composed of optimal solutions to subproblems) and the property of overlapping subproblems (i.e., the overall problem can be broken down into several subproblems of the same type whose solutions can be composed to obtain an overall solution). Clearly, both properties are not fulfilled in general in the online versions of combinatorial optimization problems. Moreover, many online problems involve time considerations such as input element release times. Therefore, it is impossible to subdivide the overall problem into several discrete stages. To apply the theory of dynamic programming, we therefore need a fixed time horizon in order to determine an optimal solution. In online optimization, the end of the input sequence is not known and decisions are made in an exclusively forward-moving rolling time horizon. The time horizon $T$ in dynamic programming is the number of periods for which the planning shall be conducted, and between periods 1 and $T$ all possible realizations in the respective periods are considered. Conceptually different, lookahead in online optimization only takes into account the actual upcoming realizations (known due to some lookahead device) and not all possible realizations. Finally, the goal in dynamic programming is different than in online optimization: in dynamic programming we are looking for an optimal strategy for given horizon, state space, action space, state transition and reward function which all are known in advance; in online optimization, we usually already have a strategy in form of an algorithm and want to check its behavior in the online setting. Nonetheless, it is possible to emulate the behavior of an online algorithm by means of a Markov chain (cf. also [22]). However, this approach is very unhandy and leads to computational issues even for small problem instances. We also note that the setup of a Markov decision process [49] is different from online optimization: state transitions occur probabilistically once a control action has been chosen, whereas in our setting they occur deterministically based on an algorithm's deterministic decision. Markov decision processes are used as a modeling formalism to determine an optimal strategy, i.e., the decisions of an optimal algorithm with respect to some expected objective value, using dynamic programming. Stochastic assumptions concerning transition probabilities depending on the control action are given a priori: $p(s, a, s')$ with $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$ is the probability that the successor state of $s$ is $s'$ if action $a$ is chosen. In contrast to this, our analysis merely intends to evaluate the quality of a given algorithm in a setting of complete nescience of stochastic information. In particular, it is not possible to find an optimal algorithm because the end of the horizon is unknown. The field of model predictive control [15] deals with finding the optimal control of complex dynamic systems. This idea is similar to that of online optimization with lookahead. However, the setting in model predictive control is relatively clear marked out by relations between dependent and independent variables in a corresponding process model. This is also why this technique is mainly used in the context of process industries, but not in the field of combinatorial online optimization. Robust optimization [11] does not rely on probability distributions but on a given range of possible values for uncertain factors. The goal is to construct a solution which is feasible for all possible realizations and exhibits optimality in some robustness-related sense. In online optimization with lookahead, there is no need for forecasts and probabilities, and subproblems are computationally easy because of their limited size. From this discussion we see that the concept of “lookahead” is seen from quite a number of different perspectives.