INFORMS TutORials in Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Online Optimization—An Introduction

Patrick JailletMichael R. Wagner

TUTORIALS IN OPERATIONS RESEARCH
INFORMS 2010

 | ISBN 978-0-9843378-0-4
DOI 10.1287/educ.1100.0072

Online Optimization—An Introduction

Patrick Jaillet
Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, and Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, jaillet@mit.edu

Michael R. Wagner
Graduate Business Programs, Saint Mary's College of California, Moraga, California 94556, mrw2@stmarys-ca.edu

Abstract It has been widely recognized (and in some cases for quite a long time) within various communities that there are many situations in which present actions must be made and resources allocated with incomplete knowledge of the future. The difficulty in these situations is that we may have to make our decision based only on the past and the current task we have to perform. It is not even clear how to measure the quality of a proposed decision strategy. The approach usually taken is to devise some probabilistic model of the future and act on this basis. This was the starting point of the theory of Markov decision processes. Other recent approaches—all under the same name, robust optimization—have been proposed, where uncertainty is characterized by set membership, rather than distributions. Online optimization is a different approach, popularized within computer science, and aims at comparing the performance of a strategy that operates with no knowledge of the future (online) with the performance of an optimal strategy that has complete knowledge of the future (offline). In this tutorial, we are proposing to give an overview of the state of the art of this approach—its applications, tools, and techniques—highlighting the relevance to operations research and management science.

Keywords online optimization; uncertainty; competitive analysis; real time

1. Introduction

Deterministic optimization techniques have proven to be powerful tools in tackling complicated decision-making problems in various application domains. Scheduling workforces, pricing airline tickets, managing production systems, and designing engineering projects are just a small sample of the myriad industrial applications that have benefited from optimization techniques such as linear, nonlinear, and integer programming, just to name a few.

Despite the many successes, there are limitations to deterministic optimization techniques. A prominent underlying assumption is frequently violated in practice, namely, that problem data are known exactly. One can resort to optimization techniques that account for uncertainty in the problem data. For example, stochastic optimization, where probabilistic distributions characterize uncertain problem parameters, is an approach that has been applied widely.

Although more adept at handling realistic optimization problems, stochastic optimization also has its drawbacks. The solutions output by stochastic optimization techniques are highly dependent on the distribution utilized to characterize the problem's uncertainty. It has also been noticed in many disciplines (e.g., operations, marketing, engineering, etc.) that the distributions utilized in these methods rarely match actual data. One of the reasons for this discrepancy is that, usually, simple distributions are applied so that the optimization approach is tractable; distributions that model real data well result in stochastic optimization problems that are very difficult to solve.

Therefore, in recent years, other approaches have appeared in the literature to model uncertainty in optimization problems. A methodology that is currently popular is robust optimization, where uncertainty is characterized by set membership, rather than distributions. Although arguably more robust than stochastic optimization, the solution approach again depends heavily on the structure of the membership sets.

In this tutorial we present an approach that is not at all dependent on a priori assumptions about the structure of the problem data uncertainty. Simply stated, an online optimization approach assumes that there is uncertainty in the problem data, but makes no assumption whatsoever about the structure of the uncertainty; it is only assumed that the problem data are revealed incrementally. In particular, no distributions or sets are ever defined that characterize the uncertainty.

The objective of the tutorial is to provide a survey of the latest developments of the online optimization methodology, and to cover a variety of applications of online optimization methods in the domain of operations research and management science (ORMS).

In §2 we give an introduction to the online optimization methodology. In §3 we discuss a number of applications that are appropriately modeled as an online optimization problem. In §4 we take an example and provide a more detailed description of results obtained on online vehicle routing problems. Finally, in §5, we discuss additional sources where the reader can learn more about online optimization, including a summary of a forthcoming new book entirely devoted to the topic.

2. Online Optimization Methodology

Because we have just stated that the online optimization methodology does not characterize the problem data uncertainty, the reader might wonder what can be done. A natural goal is to create optimization *strategies* that perform "relatively well," regardless of what data are actually realized. The relativity is measured with respect to another optimization strategy, for the same problem, that knows all the problem data *with certainty* a priori. Before we discuss the specifics of comparing strategies, we first give some mathematical structure to an online optimization problem.

2.1. Incrementally Revealed Problem Data

To make the above ideas more precise, we need to introduce some concepts and notation pertinent to the online optimization methodology. As mentioned previously, an online problem is one where the problem data are revealed incrementally. Decisions can and usually must be made before the entire problem instance is revealed. Any algorithm that gives a solution to an online problem is denoted an online algorithm. An offline algorithm solves the same problem but is given all of the problem data a priori; clearly, the offline algorithm will perform better than its online counterpart.

Let I denote a complete instance of an online optimization problem. We partition I into a sequence of partial problem instances, hereafter denoted *requests* σ_i: $I = (\sigma_1, \sigma_2, \ldots, \sigma_n)$. These requests are revealed incrementally and collectively define the problem instance. In general, the number of requests n is not known a priori. There are two main frameworks for revealing these requests, namely, a sequential model and a dynamic model.

In the *sequential model*, when σ_i is revealed, a decision by the online algorithm must be made before σ_{i+1} is revealed. In the *dynamic model*, the requests are revealed dynamically over time, irrespective of the actions of the online algorithm. The time at which a request is revealed is (usually) denoted as the request's *release date*. Note that time is an important dimension in the dynamic model, whereas it is not a necessary component in the sequential model.

2.2. Competitive Analysis and the Competitive Ratio

We now elaborate on our earlier goal of designing an optimization strategy that performs relatively well for any realization of the problem data. More specifically, we describe a framework, denoted *competitive analysis*, for evaluating the quality of online algorithms. For simplicity, we focus on minimization problems, though we do outline the necessary changes for maximization problems. Important concepts in the analysis of online algorithms are the notions of competitiveness and the competitive ratio. An online minimization algorithm is said to be r-competitive ($r \geq 1$) if, given any instance of the problem, the cost of the solution given by the online algorithm is no more than r multiplied by that of an optimal offline algorithm:

$$\mathrm{Cost}_{\mathrm{online}}(I) \leq r\ \mathrm{Cost}_{\mathrm{optimal}}(I), \quad \text{for all } \textit{problem instances } I.$$

The infimum over all r such that an online algorithm is r-competitive is called the competitive ratio of the online minimization algorithm. An online algorithm is said to be best possible if there does not exist another online algorithm with a strictly smaller competitive ratio. More generally, we allow for an additive positive constant c in the definition of competitiveness, as follows; however, most results will set $c = 0$.

$$\mathrm{Cost}_{\mathrm{online}}(I) \leq r\ \mathrm{Cost}_{\mathrm{optimal}}(I) + c, \quad \text{for all } \textit{problem instances } I.$$

Considering a maximization problem, an online algorithm is said to be r-competitive ($0 \leq r \leq 1$) if, given any instance of the problem, the cost of the solution given by the online algorithm is no less than r multiplied by that of an optimal offline algorithm:

$$\mathrm{Cost}_{\mathrm{online}}(I) \geq r\ \mathrm{Cost}_{\mathrm{optimal}}(I), \quad \text{for all } \textit{problem instances } I.$$

The supremum over all r such that an online algorithm is r-competitive is called the competitive ratio of the online maximization algorithm. The notion of a best possible maximizing online algorithm is defined similarly to that of a minimizing algorithm.

Returning to minimization problems, we may also define asymptotic competitiveness when there is a natural notion of the size of the problem. Let $n \in \mathbb{Z}$ index the size of the problem instance; symbolically, I_n is a problem instance of size n. We say that an online algorithm is asymptotically r-competitive if there exists a $n_0 > 0$ such that for all $n \geq n_0$, $\mathrm{Cost}_{\mathrm{online}}(I_n) \leq r\ \mathrm{Cost}_{\mathrm{optimal}}(I_n)$, for all *problem instances* I_n. We finally mention that the competitive ratio is related to a well-known decision analytic measure. In particular, for an online maximization problem, the competitive ratio is equivalent to the *minimax regret* criteria, applied to the logarithm of the objective function; see Borodin and El-Yaniv [15] for a more detailed discussion of the connection between decision and competitive analyses.

2.2.1. Randomized Online Algorithms. We can also consider randomized online algorithms. A randomized algorithm is a probabilistic distribution over a set of deterministic algorithms. For example, two runs of a randomized algorithm on the same instance can result in two different outputs.

Competitive analysis is slightly more complicated in the context of randomized online algorithms. Implicit in the definition of the deterministic competitive ratio was the notion of an *offline adversary* that gives the online algorithm a worst possible instance. In other words, an offline adversary knows all the details of an online algorithm and gives an instance that induces the competitive ratio. When dealing with randomized online algorithms, the definitions of the competitive ratio and competitiveness depend on how much the adversary knows about the randomization. We now compare the online cost with an *adversarial* cost.

We mention three adversaries that are traditionally used in the analysis of randomized online algorithms. These adversaries are denoted the *oblivious*, *adaptive-online*, and *adaptive-offline* adversaries. Mirroring the current research trends in online optimization, we exclusively utilize the oblivious adversary.

The oblivious adversary knows the randomized online algorithm as well as the distribution that underlies the randomness. However, the oblivious adversary has no information about the *realizations* of the distribution that drives the randomized algorithm. Therefore, the oblivious adversary creates a problem instance a priori without seeing any of the probabilistic actions of the online algorithm. In this manner, worst-case problem instances can be "averaged out" by the randomized algorithm; this is the key motivation for using randomization. The oblivious adversarial cost is deterministic and equal to the optimal offline cost. The definition of competitiveness is modified by simply replacing the cost of the online algorithm with its *expected cost*:

$$\mathbb{E}[\,\mathrm{Cost}_{\mathrm{online}}(I)] \leq r\ \mathrm{Cost}_{\mathrm{optimal}}(I), \quad \text{for all } \textit{problem instances } I.$$

2.3. Theoretically and Practically Efficient Online Algorithms

The study of online algorithms has many similarities to the study of approximation algorithms. The design of online algorithms addresses the situation where there is a limitation on information. Approximation algorithm design addresses the situation where there is a limitation on computational power. In both cases, a worst-case measure is usually applied for evaluating algorithms, namely, the competitive ratio and the approximation ratio, respectively.

We can consider both pure online problems, where we focus solely on the lack of information, as well as hybrid problems, where we are concerned with both the lack of information and the lack of computational power. For the cases where we are concerned about computational and efficiency issues, we consider both theoretically as well as practically efficient online algorithms.

We define theoretically efficient algorithms, in the usual sense, as those whose runtimes are polynomial in the size of the problem data. In this case, the competitive ratios of polynomial-time online algorithms are also approximation ratios.

We define practically efficient, or real-time, online algorithms as those whose runtimes are fast enough to be employed in practice. It is not necessary that the solutions are optimal, nor is it required that the runtimes be polynomial in the problem size. As long as a solution can be calculated in a practically acceptable runtime, we consider the online algorithm practically efficient.

3. Applications of Online Optimization

In this section, we briefly discuss a number of applications of online and real-time optimization. We give examples from routing, computer science, scheduling, revenue management, and auctions.

FedEx and other courier companies offer many real-time services. For example, FedEx provides same day pick-up and delivery. In many cases, there exist data for customers that request these services and forecasts can be created to help in designing routes. However, in the case where FedEx is beginning service in a new area, online and real-time optimization are more appropriate approaches until data can be collected about the customer base. Another application is real-time fleet management, where real-time information is used to manage a fleet of vehicles. A third example comes from a rather new industry: jet taxis. A jet taxi is a small jet that serves regional airports; customers can request transportation from one airport to another in real time. Because this is a new industry, data are not yet available to create forecasts of customer demand, so online and real-time optimization are again appropriate for analyzing these problems. Management of a fleet of taxis can also benefit from online analysis.

There is also a number of applications of online optimization in computer science. For example, paging in a memory system is one of the first applications of online optimization. Searching an unknown domain for a prize is also popular; this latter problem has applications in robotics. Routing and call admission in communications networks are also popular areas of research in online optimization.

Considering online scheduling, there are many applications in manufacturing. For a very interesting example, consider the case study presented in Ascheuer et al. [6]. Applications in warehousing also exist; consider the real-time orders that Amazon.com receives and must fill quickly (e.g., customers requesting next-day delivery).

In the airline industry, the arrival process of customers who purchase tickets is modeled well by the online framework. Although in many markets data do exist to create accurate forecasts for these customers, and a stochastic optimization approach is quite valid, in other (perhaps new) markets, a sufficient amount of data is not available, and the online optimization approach is more appropriate.

Finally, online (in the more well-known definition of the word) auctions, such as those on eBay, provide another application domain where online optimization is appropriate. The arrival of bidding customers for an auction can be appropriately modeled using the online framework. Although it is perhaps true that a sufficient amount of data exists for characterizing customer demand for many common products, it is for those less common, though perhaps no less important, products, whose data are more scarce, that online optimization is more appropriate.

4. An In-Depth Example: Online Traveling Salesman Problems and Variants

4.1. The Online TSP

The literature for the traveling salesman problem (TSP) is vast. The interested reader is referred to the books by Lawler et al. [30] and Korte and Vygen [28] for comprehensive coverage of results concerning the classical TSP.

Research concerning online versions of the TSP is more recent but has been growing steadily. Kalyanasundaram and Pruhs [27] examine a unique version where new cities are revealed locally during the traversal of a tour (i.e., an arrival at a city reveals any adjacent cities that must also be visited). Angelelli et al. [3, 4] study related online routing problems in a multiperiod setting. Bent and Van Hentenryck [12] have looked at online stochastic optimization techniques (e.g. scenario-based approaches) for addressing dynamic online routing problems; see their book (Van Hentenryck and Bent [33]) for more references and applications of these approaches.

More closely related to our tutorial is the stream of works that started with the paper by Ausiello et al. [10]. In this paper, the authors study the online version of the TSP with release dates. The formal definition of the online and offline version of the problem is as follows.

The TSP with release dates:

Instance: We are given a metric space $\mathcal{M}$ with a given origin o and a distance metric $d(\cdot,\cdot)$. A series of n requests $(l_i, r_i)_{1\le i\le n}$, where $l_i \in \mathcal{M}$ is the location (point in metric space) and $r_i \in \mathbb{R}_+$ the release date (first time after which service can be done) of request i. The problem begins at time 0; the server is initially idle at the origin (initial state), can travel at unit speed (when not idle), and eventually needs to be idle at the origin at the end (final state). The earliest time the server reaches this final state will be called the makespan.

Offline version: The number of requests n is known to the offline server. All requests $(l_i, r_i)_{1\le i\le n}$ are revealed to the offline server at time 0.

Online version: The number of requests n is not known to the online server. Requests are revealed to the online server at their release dates $r_i \ge 0$; assume $r_1 \le r_2 \cdots \le r_n$.

Objective: In both cases, minimize {the makespan to serve all accepted requests}.

We will hereafter refer to the online version of the TSP with release dates as the "online TSP." Ausiello et al. [10] provide a best possible online algorithm with a competitive ratio of 2 for the online TSP. Let us illustrate how such a result is obtained. First let us provide a simple proof (due to Lipmann [32]) of the following lower bound on the competitive ratio of any online algorithms for the online TSP:

Theorem 1 (Ausiello et al. [10], Lipman [32]). *Any c-competitive online algorithm has $c \geq 2$.*

Proof. The metric space is a graph with vertex set $X = \{o, 1, 2, \ldots, n\}$ with the origin o, and with distance metric $d(o,i) = 1$ and $d(i,j) = 2$ for all $i, j \in X \backslash \{o\}$.

At time 0 there is a request in each of the n points in $X \backslash \{o\}$. If the online algorithm serves the request in point i at time t with $t < 2n - 1 - \epsilon$, then at time $t + \epsilon$, a new request in point i is presented.

In this way, at time $2n - 1$, the online algorithm still has to serve requests in all n points. The makespan of the online algorithm will then be greater than or equal to $2n - 1 + 2n - 1 = 4n - 2$, whereas the makespan of the optimal offline algorithm will be $2n$, yielding c arbitrarily close to 2 if we take n large enough. $\square$

Consider now the following online algorithm.

Algorithm 1 (Plan-at-Home (PAH))

(1) Whenever the salesman is at the origin, it starts to follow an optimal route that serves all the revealed requests yet to be served and goes back to the origin.

(2) If at some time a new request is presented at point l, then the salesman takes one of two actions depending on its current position p:

(2a) If $d(l,o) > d(p,o)$, then the salesman immediately goes back to the origin where it then reenters a Case 1 situation.

(2b) If $d(l,o) \leq d(p,o)$, then the salesman ignores it until it arrives at the origin, where again it reenters Case 1.

Theorem 2 (Ausiello et al. [10]). *The competitive ratio of PAH is* 2.

Proof. For simplicity, let r denote the time of the final request and l the position of this request. Let $p(t)$ denote the location of the salesman at time t. Finally, let Z^* denote the optimal cost of the offline problem, and L_{TSP} the optimal offline cost when all release dates are zero. We show that in each of the Cases 1, 2a, and 2b, PAH is 2-competitive.

Case 1. PAH is at the origin at time r. Then it starts an exact tour that serves all the unserved requests. The time needed by PAH is at most $r + L_{\text{TSP}} \leq 2Z^*$.

Case 2a. We have that $d(o,l) > d(o,p(r))$. PAH returns to the origin, where it will arrive before time $r + d(o,l)$. After this, PAH computes and follows an exact tour through all the unserved requests. Therefore, the online cost is at most $r + d(o,p(r)) + L_{\text{TSP}} < r + d(o,l) + L_{\text{TSP}}$. Noticing that $r + d(o,l) \leq Z^*$, we have that the online cost is at most $2Z^*$.

Case 2b. We have that $d(o,l) \leq d(o,p(r))$. Suppose PAH is following a route $\mathcal{R}$ that had been computed the last time it was at the origin. Note that $\mathcal{R} \leq L_{\text{TSP}} \leq Z^*$. Let $\mathcal{Q}$ be the set of requests temporarily ignored because the last time PAH was at the origin; $\mathcal{Q}$ is not empty because it contains l. Let l_q be the location of the first request in $\mathcal{Q}$ served by the offline algorithm, and let r_q be the time at which l_q was released. Let $\mathcal{P}_\mathcal{Q}$ be the shortest path that starts at l_q, visits all the cities in $\mathcal{Q}$, and ends at the origin. Clearly, $Z^* \geq r_q + \mathcal{P}_\mathcal{Q}$ and $Z^* \geq d(o,l_q) + \mathcal{P}_\mathcal{Q}$.

At time r_q, the distance that PAH still has to travel on the route $\mathcal{R}$ before arriving at the origin is at most $\mathcal{R} - d(o,l_q)$, because $d(o,p(r_q)) \geq d(o,l_q)$ implies that PAH has traveled on the route $\mathcal{R}$ a distance not less than $d(o,l_q)$. Therefore, it will arrive at the origin before time $r_q + \mathcal{R} - d(o, l_q)$. After that it will follow an exact tour $\mathcal{T}_{\mathcal{Q}}$ that covers the set $\mathcal{Q}$. Hence, the completion time will be at most $r_q + \mathcal{R} - d(o, l_q) + \mathcal{T}_{\mathcal{Q}}$. Because $\mathcal{T}_{\mathcal{Q}} \le d(0, l_q) + \mathcal{P}_{\mathcal{Q}}$, we have that the online cost is at most

$$r_q + \mathcal{R} - d(o, l_q) + d(0, l_q) + \mathcal{P}_{\mathcal{Q}} = (r_q + \mathcal{P}_{\mathcal{Q}}) + \mathcal{R} \le Z^* + Z^* = 2Z^*. \quad \square$$

4.2. Overview of Other Results on the Online TSP and Variants

In Ausiello et al. [10], the authors also provide a polynomial-time online algorithm, for general metric spaces, which is 3-competitive. Subsequently, the paper by Ascheuer et al. [7] implies the existence of a polynomial-time algorithm, for general metric spaces, which is 2.65-competitive, as well as a $(2+\epsilon)$-competitive $(\epsilon > 0)$ algorithm for Euclidean spaces. Lipmann [31] develops an optimal online algorithm for the real line, with a competitive ratio of 1.64. Blom et al. [13] give an optimal online algorithm for the nonnegative real line, with a competitive ratio of $\frac{3}{2}$, and also consider different adversarial algorithms in the definition of the competitive ratio. Jaillet and Wagner [24] introduce the notion of a disclosure date, which is a form of advanced notice for the online salesman, and quantify the improvement in competitive ratios as a function of the advanced notice. A similar approach was taken by Allulli et al. [2] in the form of a *lookahead*.

There has also been work on generalizing the basic online TSP framework. The paper by Feuerstein and Stougie [19] considers the online dial-a-ride problem, where each city is replaced by an origin–destination pair. The authors consider both the uncapacitated case, giving a best possible 2-competitive algorithm, and the capacitated case, giving a 2.5-competitive algorithm. The previously cited paper by Ascheuer et al. [7] also gives a 2-competitive online algorithm and a $(1+\sqrt{1+8\rho})/2$-competitive polynomial-time online algorithm for the uncapacitated online dial-a-ride problem (ρ being the approximation ratio of a simpler but related offline problem). Their algorithm is generalizable to the case where there are multiple servers with capacities; this generalization is also 2-competitive. Jaillet and Wagner [25] consider the (1) online TSP with precedence and capacity constraints and the (2) online TSP with m salesmen. For both problems, they propose 2-competitive online algorithms (optimal in case of the m-salesmen problem), consider polynomial-time online algorithms, and then consider resource augmentation, where the online servers are given additional resources to offset the powerful offline adversary advantage. Finally, they study online algorithms from an asymptotic point of view and show that, under general stochastic structures for the problem data, *unknown and unused by the online player*, the online algorithms are almost surely asymptotically optimal.

Finally, there has been recent work dealing with online routing problems that do not require the server to visit every revealed request. Ausiello et al. [9] analyze the online quota TSP, where each city to be visited has a weight associated with it, and the server is given the task to find the shortest subtour through cities in such a way as to collect a given quota of weights by visiting the chosen cities. They present an optimal 2-competitive algorithm for a general metric space. Wagner [34] considers the case where whenever a new request arrives, the online algorithm has the option of rejecting the request and paying a given penalty. In the model considered, there are, however, no release dates; rather, the online algorithm is presented a series of requests, and must decide to accept or reject each request before the next one is revealed. The overall objective is to minimize the tour length through the accepted requests plus the sum of the penalties associated with the rejected requests. The paper presents an optimal 2-competitive algorithm for this problem on the nonnegative half-line. Ausiello et al. [8] provide a competitive analysis of the "prize-collecting TSP," a generalization of the quota problem where penalties for not visiting cities are also included, beyond meeting a given quota. They provide a 7/3-competitive algorithm and a lower bound on any competitive ratios of 2 for a general metric space, and refer to a 2-competitive algorithm and a lower bound of 1.89 on the nonnegative real line. More generally, assuming a ρ-approximation algorithm for the offline problem, they show that their online algorithm is a $(2\rho + \rho/(1 + 2/\rho))$-competitive polynomial-time algorithm. In Jaillet and Lu [22], the authors consider the "TSP with flexible service," a special case of the online prize-collecting TSP with no quotas. On the half-line and real line, they provide optimal 2-competitive online algorithms, and a c-competitive online algorithm, where $c = (\sqrt{17} + 5)/4 \approx 2.28$ for the general metric space.

Finally, in Jaillet and Lu [23], the same authors generalize and improve these results to the problem and consider two online versions of these problems. In the first online version of these problems, they assume that the server's decision to accept or reject a request can be made any time after its release date. In the second version, they assume that the server's decision to accept or reject a request must be made exactly at its release date. For the basic version of the problem in a general metric space, they construct an optimal 2-competitive online algorithm, improving the 7/3-competitive online algorithm of Ausiello et al. [8] and the 2.28-competitive online algorithm of Jaillet and Lu [22]. For the real-time version, they first prove the optimality of a 2.5-competitive polynomial-time online algorithm on the nonnegative real line. They then provide a 3-competitive online algorithm on the real line and prove a general lower bound of 2.64 and a tighter lower bound of 2.73 among a restricted family of online algorithms. Finally, they show that there cannot be any finite c-competitive online algorithm on a general metric space. They show a $\Omega(\sqrt{\ln n})$ lower bound on any competitive ratios, and describe an asymptotically optimal $O(\sqrt{\ln n})$-competitive online algorithm.

5. Where to Learn More About Online Optimization

As we indicated before, online optimization originated and has been successfully applied for many years in the computer science community. There are a few books on this topic currently in existence. Borodin and El-Yaniv [15] is a textbook on online optimization, but again the focus is primarily on computer science applications. Buchbinder and Naor [17] is a recent monograph that utilizes a primal-dual approach to analyze online problems that can be formulated as either a set-cover or packing type of problem. Van Hentenryck and Bent [33] is another recent monograph that focuses on online *stochastic* combinatorial problems, where probabilistic information supplements the purely online optimization framework. There have also been numerous books that are compilations of online optimization research: Fiat and Woeginger [20] is a collection of contributed chapters, focused primarily on computer science applications, that represented the state of the art in online optimization, when it was released in 1998. Similarly, Bampis et al. [11] is a more recent (2006) compilation of recent results in online optimization, with minimal coverage of problems in management science. On the more practical side, Grötschel et al. [21] review contemporary online optimization in large-scale systems. Finally, there are the yearly proceedings of the Workshop on Approximation and Online Algorithms that can also be considered books on online optimization, but again there is minimal coverage of problems in management science. There have also been numerous attempts to increase the visibility of online optimization, such as the survey in Albers [1], an *Optima* (newsletter of the Mathematical Programming Society) article (Ascheuer et al. [5]), as well as an Association for Computing Machinery *SIGACT News* article (Chrobak [18]). Additionally, there is a repository on the Internet that lists a large number of online optimization references: http://www.cs.ucr.edu/~marek/pubs/online.bib. Finally, online optimization has recently generated a number of dissertations: Bonifaci [14], Buchbinder [16], Krumke [29], Lipmann [32], and Wagner [35].

In recent years, numerous researchers have pioneered the application of online optimization toward problems in ORMS. Unfortunately, there is not a centralized resource for those interested in applying online optimization to ORMS problems. In the remaining part of this section, we give a summary of the content of a new book (Jaillet and Wagner [26]), which we hope can fill this gap.

The book is designed to provide a comprehensive introduction to online optimization for those in the fields of operations research and the management sciences. Because it is a research monograph, there is a heavy focus on theory, conducted by both the authors and by other researchers. We have strived to be contemporary in our selection of topics so that the book is cutting edge. In fact, a large portion of the research presented in the book is currently in press with top academic journals.

The book is organized into three parts. Part I (Classic Online Optimization) consists of Chapters 2–5 and focuses on the pure application of online optimization, without resorting to relaxations of competitive analysis; the results in this part can be considered overly conservative. However, the competitive ratio results can be interpreted as ironclad performance guarantees in practical settings. Part I focuses heavily on theory and proving competitive ratios.

Part II (Relaxations of Classic Online Optimization) consists of Chapters 6–8. Chapters 6 and 7 introduce various approaches, respectively, deterministic and probabilistic, that relax the conservatism of competitive analysis. These ideas are intended to improve the applicability of the online optimization approach to practitioners who are not overly cautious. Slightly out of place, Chapter 8 gives a detailed presentation of alternative approaches to optimization under uncertainty. In particular, comparisons between online optimization and other approaches are discussed.

Part III (Practical Online Optimization) consists of Chapters 9–11 and discusses the most practical issues in online optimization and the closely related discipline of real-time optimization. The design of practical online and real-time algorithms is emphasized in this part. Computational studies of a selection of online algorithms discussed in Parts I and II are also presented.

We next give a summary of key chapters of the book, beyond the introductory chapter (Chapter 1).

In Chapter 2, we detail the application of the online optimization framework to problems in routing. We start with an online version of the TSP, where the number of cities and their locations are not known. We then show how the analysis for this simple case can be extended to more complicated, yet realistic, routing problems that incorporate fleets of vehicles, physical capacities of vehicles, and even precedence constraints. We also consider different variants of the online TSP where quotas must be fulfilled and distances are asymmetric. Finally we consider an online version of the traveling repairman problem, where a different objective function is utilized. We consider both pure online as well as polynomial-time online algorithms in this chapter.

In Chapter 3, we present online strategies for searching optimization problems, where a prize must be located in an unknown domain. We first consider the case where a single searcher must find a prize. We then present the case where multiple searchers are looking for an item. We show that minimizing time and minimizing distance result in markedly different best possible online strategies, whereas for a single searcher, these objectives are equivalent. All strategies considered in this chapter are polynomial time.

In Chapter 4, we present online algorithms for a variety of machine scheduling problems, which is an area of ORMS where the online optimization methodology has probably made the most penetration. We consider both single and parallel machine environments under both the makespan and min-sum objectives. We also consider both preemptive and non-preemptive conditions, as well as deterministic and randomized (under the oblivious adversarial model) online algorithms. What might be most interesting about this chapter is that we present online machine scheduling problems from both the sequential and dynamic online model points of view. All algorithms presented in this chapter are polynomial time.

In Chapter 5, we present problem domains where online optimization has had limited exposure. Specifically, these are areas where future online optimization research could very well be fruitful. We consider online versions of problems in revenue management, auction theory, matching, bin packing, graph coloring, and telecommunication networks.

In Chapter 6, we discuss deterministic relaxations of competitive analysis. Because the competitive ratio is a worst-case measure, many commentaries have legitimately claimed that online strategies are too conservative. We introduce a number of deterministic relaxations that are realistic and are successful in reducing the conservatism of competitive analysis.

In Chapter 7, we also introduce relaxations to competitive analysis; however, these relaxations are probabilistic in nature. Though probabilistic distributions are introduced, the problems are still inherently online problems and do not fall in the domain of stochastic optimization.

In Chapter 8, we detail alternative approaches to optimization under uncertainty. In particular, we give the basic ideas behind chance-constrained programming, stochastic optimization with recourse, dynamic programming and optimal control theory, robust optimization, and a priori optimization. We also emphasize the differences, both good and bad, when comparing online optimization with each of these alternative approaches.

In Chapter 9, we focus on real-time optimization algorithms, where theoretical properties are deemphasized, and practicality is the focus. In particular, we detail a real-time optimization approach to a practical pick-up and delivery problem.

In Chapter 10, we present a practical study of real-time auctions and pricing in the transportation sector. Again, theory is not emphasized, and intuition is built through computational experiments.

In Chapter 11, we present computational studies of theoretical algorithms analyzed in previous chapters. In particular, we present studies of a selection of routing and machine scheduling algorithms.

Finally, in the appendix, we give technical details that would be out of place in the main chapters. In particular, we discuss asymptotic notation, polynomial-time algorithms, and different notions of stochastic convergence. We also discuss technical details that are more specific to online optimization, namely, different adversarial models for randomized online algorithms and the Yao principle.

Acknowledgments

This tutorial is part of a series of collaborative works which began with the Patrick Jaillet's dissertation.

References

[1] S. Albers. Online algorithms: A survey. *Mathematical Programming* 97(1–2):3–26, 2003.

[2] L. Allulli, G. Ausiello, V. Bonifaci, and L. Laura. On the power of lookahead in on-line server routing problems. *Theoretical Computer Science* 408(2–3):116–128, 2008.

[3] E. Angelelli, M. Savelsbergh, and M. Speranza. Competitive analysis for dynamic multi-period uncapacitated routing problems. *Networks* 49(4):308–317, 2007.

[4] E. Angelelli, M. Savelsbergh, and M. Speranza. Competitive analysis of a dispatch policy for a dynamic multi-period routing problem. *Operations Research Letters* 35(6):713–721, 2007.

[5] N. Ascheuer, M. Gröetschel, and J. Rambau. Combinatorial online optimization in practice. *Optima* 57:1–6, 1998.

[6] S. Ascheuer, M. Gröetschel, J. Rambau, and N. Kamin. Combinatorial online optimization in practice. *Optima, Mathematical Programming Society Newsletter* 57:1–6, 1998.

[7] N. Ascheuer, S. Krumke, and J. Rambau. Online dial-a-ride problems: Minimizing the completion time. *Proceedings of the 17th International Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science*, Vol. 1770. Springer, Berlin, 639–650, 2000.

[8] G. Ausiello, V. Bonifaci, and L. Laura. The on-line prize-collecting traveling salesman problem. *Information Processing Letters* 107(6): 199–204, 2008.

[9] G. Ausiello, M. Demange, L. Laura, and V. Paschos. Algorithms for the on-line quota traveling salesman problem. *Information Processing Letters* 92(2):89–94, 2004.

[10] G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, and M. Talamo. Algorithms for the on-line travelling salesman. *Algorithmica* 29(4): 560–581, 2001.

[11] E. Bampis, K. Jansen, and C. Kenyon. *Efficient Approximation and Online Algorithms: Recent Progress on Classical Combinatorial Optimization Problems and New Applications*. Springer, Berlin, 2006.

[12] R. Bent and P. Van Hentenryck. Scenario based planning for partially dynamic vehicle routing problems with stochastic customers. *Operations Research* 52(6):977–987, 2004.

[13] M. Blom, S. O. Krumke, W. E. de Paepe, and L. Stougie. The online TSP against fair adversaries. *INFORMS Journal on Computing* 13(2):138–148, 2001.

[14] V. Bonifaci. Models and algorithms for online server routing. Ph.D. thesis, Universita di Roma La Sapienza, Rome, 2007.

[15] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, Cambridge, UK, 1998.

[16] N. Buchbinder. The design of competitive online algorithms via a primal-dual approach. Ph.D. thesis, Technion, Haifa, Israel, 2008.

[17] N. Buchbinder and J. Naor. *The Design of Competitive Online Algorithms via a Primal-Dual Approach*. Now Publishers Inc., Delft, The Netherlands, 2009.

[18] M. Chrobak. Online algorithms. *SIGACT News* 34(4):68–77, 2003.

[19] E. Feuerstein and L. Stougie. On-line single-server dial-a-ride problems. *Theoretical Computer Science* 268(1):91–105, 2001.

[20] A. Fiat and G. Woeginger, eds. *Online Algorithms: The State of the Art. Lecture Notes in Computer Science* (*State of the Art Survey*), Vol. 1442. Springer-Verlag, Berlin, 1998.

[21] M. Grötschel, S. Krumke, and J. Rambau. *Online Optimization of Large Scale Systems: State of the Art*. Springer, Berlin, 2001.

[22] P. Jaillet and X. Lu. Online traveling salesman problems with service flexibility. *Networks*. Forthcoming,

[23] P. Jaillet and X. Lu. Online traveling salesman problems with immediate decisions. Working paper, Operations Research Center, Massachusetts Institute of Technology, Cambridge, 2010.

[24] P. Jaillet and M. Wagner. Online routing problems: Value of advanced information as improved competitive ratios. *Transportation Science* 40(2):200–210, 2006.

[25] P. Jaillet and M. Wagner. Generalized online routing: New competitive ratios, resource augmentation and asymptotic analyses. *Operations Research* 56(3):745–757, 2008.

[26] P. Jaillet and M. Wagner. *Online Optimization*. International Series in Operations Research and Management Science, Springer-Verlag, Berlin, 2010.

[27] B. Kalyanasundaram and K. R. Pruhs. Constructing competitive tours from local information. *Theoretical Computer Science* 130(1):125–138, 1994.

[28] B. Korte and J. Vygen. *Combinatorial Optimization, Theory and Algorithms*, 2nd ed. Springer, Berlin, 2002.

[29] S. Krumke. Online optimization: Competitive analysis and beyond. Ph.D. thesis, Habilitationsschrift, Technische Universitat Berlin, Berlin, 2002.

[30] E. Lawler, J. Lenstra, A. H. G. Rinnooy Kan, and D. Shmoys. *The Traveling Salesman Problem, A Guided Tour of Combinatorial Optimization*. John Wiley & Sons Ltd., Chichester, UK, 1985.

[31] M. Lipmann. The online traveling salesman problem on the line. Master's thesis, University of Amsterdam, Amsterdam, 1999.

[32] M. Lipmann. On-line routing. Ph.D. thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2003.

[33] P. Van Hentenryck and R. Bent. *Online Stochastic Combinatorial Optimization*, 1st ed. MIT Press, Cambridge, MA, 2006.

[34] M. Wagner. Online TSP with rejections. Completed research note, California State University, East Bay, Hayward, 2006.

[35] M. Wagner. Online optimization in routing and scheduling. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, 2006.