Diana Poensgen

Facets of Online Optimization

Online Dial-a-Ride Problems
and
Dynamic Configuration of All-Optical Networks

Cuvillier Verlag Göttingen

Facets of Online Optimization

Online Dial-a-Ride Problems and Dynamic Configuration of All-Optical Networks

vorgelegt von
Dipl.-Math. Diana Poensgen
Berlin

Von der Fakultät II – Mathematik- und Naturwissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
Dr. rer. nat.

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Dirk Ferus
1. Berichter: Prof. Dr. Martin Grötschel
2. Berichter: Dr. Sven O. Krumke

Tag der wissenschaftlichen Aussprache: 26. August 2003

Berlin 2003
D 83

Bibliografische Information Der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.
1. Aufl. - Göttingen : Cuvillier, 2003
Zugl.: Berlin, Univ., Diss., 2003
ISBN 3-89873-919-8

Nonnenstieg 8, 37075 Göttingen
Telefon: 0551-54724-0
Telefax: 0551-54724-21
www.cuvillier.de

1. Auflage, 2003
Gedruckt auf säurefreiem Papier

ISBN 3-89873-919-8

Zusammenfassung

Die Online-Optimierung befasst sich mit Optimierungsproblemen, die unmittelbare und schnelle Entscheidungen auf Basis unvollständiger Information erfordern. In dieser Arbeit untersuchen wir zwei Klassen von Online-Optimierungsproblemen: Online-Dial-a-Ride-Probleme und die dynamische Konfiguration optischer Netze. Bei einem Online-Dial-a-Ride-Problem muss ein Bediengerät Transportaufträge ausführen, die im Verlauf der Zeit eintreffen. Typische Anwendungen sind die Steuerung von Aufzügen sowie Auslieferdienste. Die Aufgabe bei der dynamischen Konfiguration optischer Telekommunikationsnetze besteht darin, nach und nach eintreffende Verbindungsanfragen zu routen oder sie abzulehnen. Das Ziel ist es, den durch die Annahme von Anfragen erhaltenen Gesamtprofit zu maximieren. Dabei müssen spezielle Eigenschaften der optischen Technologie berücksichtigt werden.

Wir betrachten Online Dial-a-Ride Probleme mit zwei verschiedenen Zielfunktionen: der gewichteten Summe der Bedienzeiten und der maximalen Flusszeit. Für die Summe der Bedienzeiten stellen wir neue Algorithmen vor und bewerten sie mittels kompetitiver Analyse. Zusätzlich beweisen wir neue untere Schranken. Alle vorgestellten Resultate verbessern die bisher bekannten Ergebnisse. Die maximale Flusszeit ist für die Praxis besonders relevant, da sie ein Maß für die maximale Kundenunzufriedenheit darstellt. Hier zeigt sich die Schwäche der kompetitiven Analyse: Werden Online-Algorithmen mit diesem worst-case-Ansatz evaluiert, so sind sie alle beliebig schlecht. Wir präsentieren eine natürliche Modifikation der kompetitiven Analyse, die es uns erlaubt, den ersten beweisbar kompetitiven Algorithmus für Online-Dial-a-Ride-Probleme mit der maximalen Flusszeit als Zielfunktion zu entwickeln.

Für die Konfiguration optischer Netze stellen wir wir ein neues Online-Modell vor, das eine Vielzahl von Anforderungen aus der Praxis abbildet. Zudem entwickeln wir eine Reihe von praxisnahen sowie von theoretisch motivierten Algorithmen. Die zum Teil recht komplexen praxisnahen Algorithmen vergleichen wir mit bekannten Greedy-Algorithmen mittels Simulation. Dazu erzeugen wir Verbindungsanfragen für zwei existierende Topologien anhand eines fundierten Verkehrsmodells, das auf realen Daten basiert. Auf den betrachteten Probleminstanzen ist der beste Greedy-Algorithmus nur dann annähernd so gut wie der beste der neuen Algorithmen, wenn die Tie-Breaking-Regel sehr sorgfältig ausgewählt wird. Die theoretisch motivierten Algorithmen, die wir für zwei Teilproblemklassen entwickeln, bewerten wir wiederum per kompetitiver Analyse. Sie sind die ersten kompetitiven Algorithmen für die betrachteten Problemklassen und erweitern damit bekannte Resultate.

Acknowledgments

My work was made possible by the Scholarship of the *Konrad-Zuse-Zentrum für Informationstechnik Berlin*. I am greatly indebted to many friends and colleagues who contributed in various ways to this thesis. First of all, I sincerely thank my supervisors Prof. Dr. Martin Grötschel and Dr. Sven O. Krumke for their support, their guidance, and for the freedom they gave me in my research. Sven introduced me to the right people, laying the ground for and participating in numerous fruitful discussions which eventually lead to the results presented here.

I am grateful to the many people I had the opportunity to work with. My sincere thanks go to Monika Jäger, Ralf Hülsermann, Sven O. Krumke, Luigi Laura, Maarten Lipmann, Alberto Marchetti-Spaccamela, Willem de Paepe, Jörg Rambau, Leen Stougie, Andreas Tuchscherer, and Yossi Azar for letting me include our joint results in this thesis. In particular, I thank Willem and Leen for the interest, enthusiasm and effort they put in our joint research, and for all the laughter and fun together. I also wish to thank Georgio Ausiello and Alberto Marchetti-Spaccamela for giving me the opportunity to join their research group in Rome for a month. This stay was financed by the EU (Project AMORE).

I am particularly grateful to Sven O. Krumke, Jörg Rambau, Andreas Tuchscherer, and Tjark Vredeveld for inspiring discussions, helping comments, and, last but not least, the pleasant atmosphere when working together. I owe special thanks to Gunnar Klau, Jörg Rambau, Andreas Tuchscherer, Tjark Vredeveld, and Adrian Zymolka for their careful reading of the manuscript of my thesis. Their constructive remarks were most helpful. Thanks go also to Benjamin Hiller, Marc Pfetsch, and Roland Wessäly for their helpful suggestions.

Finally, I wish to thank all my colleagues of the Optimization Department at ZIB for the friendly work atmosphere. Special thanks go to my roommate Adrian who always tried his best to keep up both my spirit and my level of blood sugar.

Contents

Chapter 1

Introduction

Online Optimization

In classical optimization, all input data of a problem instance is assumed to be available at once. Many real-life problems, however, require decisions before information about the data is complete. This insight has prompted the research in *online optimization*. In an online optimization problem, decisions have to be made while parts of the data are still missing.

Many real-life problems are naturally online. Picture, for instance, your next weekend trip to a European city. You want to see the most important sights, including monuments and museums, and walk around to immerse into the city's special atmosphere. When you arrive, one of the first decisions you will need to make is whether to buy a 4-day ticket, valid on all public transport, at a price of 18 Euro, or whether to buy tickets whenever you need them. Single ride tickets have a validity of 90 minutes, and their prices range from 1.50 Euro to 2.30 Euro, depending on the length of your ride and the means of transportation you choose. So, what should you do? A 4-day ticket would be convenient, of course: you don't have to bother about buying tickets any more. But is it worth it? On the one hand, if the weather is nice, then you'll probably walk most of the time, and you could save some money now and afford another (probably too expensive) coffee on one of the main boulevards later. On the other hand, if it rains a lot, then it would be nice to take a bus ride through the city or to catch the subway to the next museum. A lot of information such as the actual weather conditions, special events you might want to attend, etc., are not known to you in advance. This forces you to make decisions under uncertainty.

Incomplete information is a feature common to many real-life problems. Online problems arising in practice include distributed data management, foreign exchange and stock trade, the control of elevators, and the routing of calls in a telecommunication network.

In online optimization, the input data is usually modeled as a sequence of *requests*

that is revealed piecewise. An online algorithm may base its decisions at any time only on the requests known to it so far. Request sequences can be classified into two types: the *sequence model* and the *time stamp model*.

In the sequence model, an online algorithm must process the requests one by one. It gets to know the next request only after it has made an irrevocable decision for the previous one. A well-known example is *paging* in a virtual memory system: a central processing unit (CPU) must decide which page to evict from its fast memory upon receiving a request for some page in its slow memory. The next request is only disclosed when the previous one has been processed, i.e., when the CPU has ensured that the requested page is in its fast memory. Time does not play any role in this model. Each decision of an online algorithm results in some gain or loss, and the objective function usually depends on the total gain or loss.

In contrast, time is decisive in the time stamp model. Here, each request has a non-negative *release time*, and an online algorithm gets to know further requests as time is progressing. The objective function usually depends on time. An online optimization problem in which new requests arrive over time is the *Online Traveling Repairman Problem*, an online variant of the famous *Traveling Salesman Problem*. In the Online Traveling Repairman Problem, a repairman has a set of jobs to do. Each job requires him to drive to a customer. While he is on his way, the repairman receives new requests and must decide how to rearrange his schedule such as to finish each job as early as possible.

Evaluation of Online Algorithms

For the comparison of online algorithms, it is necessary to have meaningful measures for assessing their quality. Several approaches have been taken to evaluate online algorithms.

In the traditional *distributional* or *stochastic* approach, a distribution over the problem instances is assumed, and the *expected* objective value of the online algorithm under consideration is computed. The major weakness of this average-case approach is that the assumed distribution is often either unrealistic, or too complex, making the computation of the expected value intractable. Another option is to compare the worst-case objective values of two algorithms. Alas, this approach is also problematic: what if all algorithms are equally bad in the worst-case? In the paging problem mentioned above, for instance, all algorithms show the same worst-case behavior: they incur a page fault in each step.

Competitive analysis tries to overcome this weakness by introducing a (hypothetical) benchmark algorithm, the *optimal offline algorithm*, that knows the given sequence in advance and can process it in an optimal way. The worst-case performance of an online algorithm is measured relative to the optimal offline algorithm:

Given a minimization problem, an online algorithm ALG is said to be *c-competitive*, if, for any problem instance σ, the objective function value of ALG on σ is at most c times the objective value of the optimal offline algorithm on σ. Since the ratio of the online and the offline objective value must be bounded by the constant c for all instances, competitive analysis is a worst-case approach. It is intended to answer the question what is lost in the worst case by the lack of complete information.

Competitive analysis has become the standard tool in online optimization. Nevertheless, it suffers from several conceptual deficiencies. For instance, it completely disregards complexity issues; an online algorithm is not required to be efficient or to make real-time decisions. Fast computation becomes important, however, when online algorithms are to be used in practice. Very frequently, decisions have to be made within a given time bound, often within seconds. Therefore, online algorithms intended for real world problems must be efficient, or at least workable.

An important instrument for the assessment of practical algorithms is simulation. Simulation experiments are indispensable to emulate and evaluate the behavior of online algorithms designed for practical use. In particular when worst-case performance bounds are not available or based on unrealistic scenarios, carefully chosen simulation runs provide valuable experimental performance guarantees. It is important to use real data for simulation whenever possible.

Other weaknesses of competitive analysis, as well as efforts to overcome them, will be addressed in Section 2.3.

Online Dial-a-Ride Problems

In the Dial-a-Ride Problem (DARP), one or several servers of given capacities and of unit speed have to transport objects between points in a metric space. Each transportation request specifies a *source* and a *destination*. The task is to design a sequence of moves for each server such that all transportation requests, also called *ride*s, are covered, the server's capacity bounds are not exceeded, and a given objective function is minimized. Moreover, unless specified otherwise, *preemption* is prohibited: once an object has been picked up, it can only be dropped at its destination. In some variants of the DARP, the servers are additionally required to eventually return to their initial position.

The class of Dial-a-Ride Problems comprises many well-studied problems in combinatorial optimization. For instance, the *Traveling Salesman Problem* is the special case of the single-server DARP with the makespan as objective function in which source and destination of each ride coincide. Also many *Vehicle Routing Problem*s can be formulated within the DARP framework. Moreover, Dial-a-Ride Problems can be used to model *scheduling problems* in which the jobs have order dependent setup costs.

We are interested in the online version of the Dial-a-Ride Problem with a single server, further referred to as the *Online Dial-a-Ride Problem* (OLDARP). In the OLDARP, transportation requests are not known beforehand but become known over time. That is, in addition to source and destination, each ride specifies a *release time*. In the online setting, the server has at no point in time any information about requests whose release time is greater than that point in time. In particular, it neither knows the total number of requests, nor the release time of the last request. Objective functions that have been considered for the OLDARP are the *makespan* (completion time of the schedule), the *latency* (weighted sum of completion times of all requests), the *average flow time* (average time in which a request remains in the system), and the *maximum flow time*. In this thesis, we present new results for the OLDARP with the latency and with the maximum flow time as objective functions.

Online Dial-a-Ride Problems occur frequently in practice, in particular in logistics. Applications are machine scheduling, field service, delivery and courier services, elevator and stacker crane control, transportation of disabled persons, and the dispatching of automobile service units, among others.

Online Call Admission in All-Optical Networks

All-optical telecommunication networks are the optical networks of the next generation. While in today's networks, signals are already transmitted as light pulses via glass fibers, but still switched electronically in intermediate nodes, new devices will shortly allow to process signals completely within the optical domain.

The *Wavelength Division Multiplexing* (WDM) technique, deployed for the first time in the early 1990ies, brought a substantial increase in transmission capacities of telecommunication networks. By installing so-called *multiplexer*s and *demultiplexer*s at the beginning and the end of a fiber, respectively, the available bandwidth of a fiber is separated into different wavelengths that can be used in parallel by different signals. Recently, new devices for the switching and the insertion/extraction of signals in optical form have been developed, the so-called *Optical Cross-Connects* and *Optical Add-Drop-Multiplexers*. They are expected to be commercially available very soon. Moreover, *wavelength converters* are being devised that enable the (optical) switching of signals from one wavelength to another. Altogether, these new devices supersede current time-consuming conversions between optics and electronics. *All-optical networks* refer to optical networks that deploy these new switches in addition to the WDM technique.

All-optical networks require new mathematical models and give rise to new problems. Their crucial difference to the networks currently in use is that a signal sent through an all-optical network remains in optical form on its whole path from start to end node. Therefore, a connection in an all-optical network is realized via a *light-*

path, which is a path in the network together with a wavelength. Resources are limited: each wavelength may only be used once per fiber; consequently, two lightpaths that use the same fiber must have different wavelengths. This crucial restriction is called *wavelength conflict constraint*. A natural online problem is the *dynamic configuration of optical networks*. In its simplest variant it can be stated as follows: new connection requests arrive over time, and an (online) algorithm has to decide for each request whether to accept or reject it (*call admission*). If the request is accepted, the algorithm must provide a lightpath to realize the required connection without violating the wavelength conflict constraint (*routing and wavelength assignment*).

Overview

This thesis is divided into two major parts: in Part I, we investigate various Online Dial-a-Ride Problems; Part II is concerned with the dynamic configuration of all-optical networks.

Preceding these two major parts is Chapter 2, which is intended as a short reference to the concepts and the basic notation used in this thesis. We give a formal introduction to online optimization and competitive analysis, including deterministic as well as randomized online algorithms. We also introduce a useful technique for obtaining lower bounds on the competitive ratio of randomized algorithms. Moreover, we discuss the weaknesses of competitive analysis and cover known modifications and extensions of it.

In Part I, we present new results for several Online Dial-a-Ride Problems. After a formal introduction to Online Dial-a-Ride Problems in Chapter 3, Chapter 4 deals with Online-Dial-a-Ride Problems with the *latency* as objective function ($\sum w_jC_j$-OLDARP). The latency is defined as the weighted sum of completion times, where the *completion time* of a request is the time when the corresponding object is dropped at its destination. We present new lower bounds on the competitive ratio of any online algorithm, both for the general $\sum w_jC_j$-OLDARP and for the special case in which each ride's source and destination coincide, the $\sum w_jC_j$-OLTSP, also known as the *Online Traveling Repairman Problem* (OLTRP). The main result of this chapter is a $(1+\sqrt{2})^2$-competitive deterministic online algorithm for the $\sum w_jC_j$-OLDARP in general metric spaces. This algorithm significantly improves previous upper bounds for both the $\sum w_jC_j$-OLDARP and the $\sum w_jC_j$-OLTSP. Moreover, a modification of the algorithm yields a new randomized upper bound.

In Chapter 5, we investigate the OLDARP with the maximum flow time $F_{\max}$ as objective function, shortly $F_{\max}$-OLDARP. Again, we also consider the special case in which each ride's source and destination coincide, the $F_{\max}$-OLTSP. Easy worst-case sequences reveal that there is no competitive online algorithm for neither problem, showing that competitive analysis fails to evaluate and distinguish online

algorithms. This motivates the search for new concepts to restrict the adversary's power in the OLDARP. As a start, we consider the concept of *fairness*, defined for Euclidean metric spaces by Blom, Krumke, de Paepe, and Stougie, see [25], and extend it to the uniform metric space. On the uniform metric space, a first-come-first-serve strategy turns out to be best possible for the $F_{\max}$-OLTSP against a fair adversary. For the more general $F_{\max}$-OLDARP on the uniform metric space, we prove that no deterministic online algorithm can be competitive against a fair adversary. We then consider the real line endowed with the Euclidean metric and show that the fairness condition is still too weak to allow for competitive algorithms, even in the $F_{\max}$-OLTSP. Therefore, we introduce a new adversary type that is subject to a stronger restriction: the *non-abusive adversary*. On the uniform metric space, the negative result for the $F_{\max}$-OLDARP against a fair adversary carries over to the non-abusive adversary. For the real line, the situation is different. One of our major results is a constant-competitive algorithm for the $F_{\max}$-OLTSP against a non-abusive adversary on the real line. This is the first competitive algorithm for the minimizing the maximum flow time on this metric space.

Part II is dedicated to the dynamic configuration of all-optical networks. In Chapter 6, we give an introduction to all-optical networks, including their technical features as well as the mathematical model used to describe them. We then report on problems involved with the dynamic configuration of all-optical networks, and we introduce an online optimization model for the so-called *Dynamic Multiclass Call Admission Problem* (DMCAP). Our model is suitable to map a variety of real-life characteristics, such as different degrees of service quality and various customer classes. Moreover, the model is suitable to map failure situations.

In Chapter 7, we present several new algorithms developed for a restricted version of the DMCAP. These *fitness-algorithms* realize the routing choice that yields the smallest decrease of the network fitness, a quantity that seeks to measure potential future profit. Our algorithms are designed for practice and evaluated by means of simulation. In particular, we compare them to known greedy strategies on realistic instances. The experimental setup and the results of the simulation runs are reported on in Section 7.4. It emerges that the best greedy algorithm performs almost as well as the best of the fitness-algorithms. Yet, traffic is comparatively evenly distributed in the instances we have considered. On more realistic scenarios, we expect our algorithms to outperform the algorithms of greedy type more significantly. Our experiments furthermore reveal that as little as a tie-breaking rule can decide about failure or success of a greedy algorithm, thereby providing important information to practitioners who often use the tie-breaking rule that is easiest to implement. The chapter closes with a discussion of our experimental set-up and an outlook on further lines of research.

Finally, Chapter 8 is dedicated to a more theoretical approach to online call admission in optical networks. We present and analyze algorithms for a restricted vari-

ant of the previously considered online call admission problem. The problem variant we consider is more general than those problems for which competitive algorithms are known to exist. Our main contribution are two competitive algorithms, one for general networks, and an improved one for trees. Moreover, we investigate the preemptive version of the problem and show how our non-preemptive algorithm for trees can be combined with techniques for preemptive call admission in non-optical line networks to obtain competitive preemptive algorithms for the line. The algorithms we present are the first competitive algorithms for preemptive online call admission in all-optical networks.

Although the algorithms presented in this chapter are not suited for immediate practical use, their assessment via competitive analysis is nevertheless helpful for the design of practical algorithms, as it helps to identify decisions that may lead to very unfavorable scenarios.

Credits

Many of the results presented in this thesis were obtained as joint work with other people. The results in the Online-Dial-a-Ride Problem with the latency as objective function, presented in Chapter 4, have been obtained together with Sven O. Krumke, Willem E. de Paepe, and Leen Stougie, see [68]. The work on Online Dial-a-Ride Problems with the maximum flow time as objective function, see Chapter 5, has been carried out jointly with Sven O. Krumke, Luigi Laura, Maarten Lipmann, Alberto Marchetti-Spaccamela, Willem E. de Paepe, and Leen Stougie, cf. [69]. Chapter 7, which reports on the practical approach to online routing problems in optical networks, is based on joint work with Ralf Hülsermann, Monika Jäger, Sven O. Krumke, Jörg Rambau, and Andreas Tuchscherer. Parts of this chapter have been published in [56]. Finally, the theoretical results in Chapter 8 about online call admission in optical networks were obtained together with Yossi Azar and Sven O. Krumke, see also [70].

I thank the aforementioned people for allowing me to include our joint results in this thesis.

Chapter 2

Preliminaries

In this chapter, we give an introduction to online optimization, competitive analysis, and simulation. We also discuss the weaknesses of competitive analysis and efforts to overcome them.

We introduce online optimization in Section 2.1. In Section 2.2, we define the main concepts of competitive analysis and briefly present *Yao's Principle*, a useful technique to obtain lower bounds on the competitive ratio of randomized online algorithms. We try to keep our presentation as precise as necessary, but at the same time intuitive enough to understand the important concepts without getting lost in formal details. A rigorous introduction to online optimization using a game-theoretic setting is given in Appendix A.

We address the weaknesses of competitive analysis and report on known modifications and extensions that have been proposed as a remedy in Section 2.3. Finally, we discuss simulation as another tool for the evaluation of online algorithms in Section 2.4.

2.1 Online Optimization

An *optimization problem* is defined by a set of *(input) instances*, a set of *valid solutions (outputs)* for each instance, an *objective function* that assigns a value to each input-output combination, and the declaration whether the objective function has to be minimized or maximized (see Section B.2 for the formal definition). In the sequel, we assume that we are given a minimization problem and use the terms objective function and *cost function* interchangeably.

An *algorithm* ALG for an optimization problem computes for each input instance I a valid solution $\mathsf{ALG}[I]$. The cost of this solution is the value assigned by the objective function to the pair $(I, \mathsf{ALG}[I])$.

Informally speaking, online optimization problems are those optimization prob-

lems in which the input is not given at once, but instead as a sequence of requests. An online algorithm must serve each request without knowing the future requests or the length of the sequence.

Definition 2.1 (Online Optimization Problem, Online Algorithm). *An* online optimization problem *is an optimization problem where the input is given as a sequence* $\sigma = r_1, \ldots, r_n$ *of* requests.

Definition 2.2 (Online Algorithm). *An* online algorithm *for an online optimization problem* Π *must give an* answer a_i *to each request* r_i *in the given sequence that is solely based on the previous requests* $r_1, \ldots, r_{i-1}$ *and its previous answers* $a_1, \ldots, a_{i-1}$*. The sequence of answers* $a_1, \ldots, a_n$ *generated by the online algorithm must have the property that* $a_1, \ldots, a_k$ *is a valid solution for the input* $r_1, \ldots, r_k$ *for each* $k = 1, \ldots, n$*.*
The output *of* ALG *on input* σ *is defined as* $\mathsf{ALG}[\sigma] := a_1, \ldots, a_n$*. The* cost *of* ALG *on input* $\sigma = r_1, \ldots, r_n$ *is defined by* $\mathsf{ALG}(\sigma) := C(\sigma, \mathsf{ALG}[\sigma])$*, where* C *is the cost function of* Π*.*
An algorithm is deterministic*, if its output is unique for each fixed input instance.*

In the sequence model, cf. Chapter 1, the algorithm must serve the current request before it is given the next one in the sequence. In the time stamp model, the requests have an associated release time at which they become known to the online algorithm. When a new request is released, the algorithm must not immediately serve it, but it must give an answer that indicates when and how it plans to serve the request. The two models are illustrated by the following examples.

Example 2.3 (k Server Problem). In the k *Server Problem*, we are given a fixed integer $k \geq 1$, and a metric space (X, d) where X is a set of points with $|X| > k$ and d is a metric on X (cf. Section B.1). An online algorithm has k servers at its disposition that are initially located in k distinct points of X, and is presented with a sequence of requests each of which specifies a point in X. Upon revelation of the next request, the algorithm must decide which of its servers to move to the requested point. The cost associated with this decision equals the distance the chosen server has to cover. The k Server Problem is an example for the sequence model. $\lhd$

Example 2.4 (Online Scheduling on Identical Machines). The following variant of *online scheduling on identical machines* is formulated using the time stamp model. An online algorithm is given a set of m identical machines, and it has to assign jobs of given duration that arrive over time to the machines. Each machine can only process one job at a time. More precisely, each request $r := (t, d) \in \mathbb{R}^{\geq 0} \times \mathbb{R}^{\geq 0}$ specifies a release time t and a duration d. We assume that the requests are ordered in non-decreasing release times and the sequence unfolds as time goes by. As soon as a new

request is released, the algorithm must assign it to a machine and give it a start time. As long as the machine has not yet started to process the job (because earlier jobs on this machine are not finished yet), this decision is revocable. The goal is to minimize the makespan, that is, the earliest time at which all jobs have been processed. ◁

We now consider randomized algorithms. For a formal introduction to the basic concepts of probability theory, see Section B.5 in the Appendix.

Definition 2.5 (Randomized Online Algorithm). *A* randomized online algorithm RALG *for an online optimization problem* Π *is a probability distribution over the set of all deterministic online algorithms for* Π.
Both the output $\mathsf{RALG}[\sigma]$ *and the cost* $\mathsf{RALG}(\sigma)$ *of* RALG *on input* σ *are random variables; its* expected cost $\mathbb{E}[\mathsf{RALG}(\sigma)]$ *on input* σ *is defined as the expected value of the random variable* $\mathsf{RALG}(\sigma)$.

2.2 Competitive Analysis

Competitive analysis provides a framework to measure the quality of online algorithms. More precisely, it seeks to answer the question what is lost in the worst-case due to lack of information. To this end, a benchmark algorithm is introduced that has complete knowledge of the request sequence in advance and can serve it in an optimal way. Such an algorithm is called an *optimal offline algorithm*. In order to be *competitive*, an online algorithm must compete well with the optimal offline algorithm on *all* input instances. Hence, competitive analysis is a worst-case approach.

Competitive analysis has been introduced formally for the first time in 1985 by Sleator and Tarjan, see [94]; the term *competitive analysis* was coined in a paper by Karlin, Manasse, Rudolph, and Sleator in 1988, cf. [60]. The method itself was already used in the 1960s, when Graham analyzed and evaluated his algorithm for list accessing, see [49] and [50]. For a more extensive survey on the history of online optimization and competitive analysis, we refer to the introductory chapter in the book by Fiat and Woeginger, cf. [42].

In this section, we define competitiveness for deterministic and randomized algorithms and provide a useful technique to derive lower bounds against randomized algorithms.

2.2.1 Deterministic Algorithms

We start with a definition of the benchmark algorithm that is required for the assessment of online algorithms.

Definition 2.6 (Optimal Offline Algorithm). *An* optimal offline algorithm OPT *for an online optimization problem* Π *is an algorithm that knows the whole input sequence* σ *in a given instance in advance and serves it at optimal cost.*
Its output $\mathsf{OPT}[\sigma]$ *on input* σ *is called the* optimal offline output*, its cost* $\mathsf{OPT}(\sigma)$ *on input* σ *the* optimal offline cost.

The optimal offline algorithm is usually not described by a set of rules but only defined via the collection of optimal outputs. In fact, the optimal offline algorithm often remains unspecified, and only the optimal offline cost is computed for specific instances, or bounded on arbitrary input sequences. The crux of the matter is that the optimal offline algorithm has the full view on the problem instance, and can choose its answer for each request such that the resulting output on the whole sequence is optimal w.r.t. the cost. The core of competitive analysis is the following definition.

Definition 2.7 (Competitive Deterministic Algorithm). *Let* Π *be an online optimization problem, and let* $c \in \mathbb{R}$*,* $c \geq 1$*. A deterministic online algorithm for* Π *is* c-competitive*, if there exists a constant* $b \geq 0$ *such that*

$$\mathsf{ALG}(\sigma) \leq c \cdot \mathsf{OPT}(\sigma) + b$$

for all input instances σ*. If* $b = 0$*, then* ALG *is said to be* strictly c-competitive.

For a *maximization* problem $\Pi_{\max}$, competitiveness is defined analogously: An online algorithm ALG for $\Pi_{\max}$ is c-competitive if there exists a constant $b \geq 0$ such that $\mathsf{ALG}(\sigma) \geq \frac{1}{c} \cdot \mathsf{OPT}(\sigma) - b$ on all input sequences σ.

Hence, an online algorithm's performance is measured in relation to that of the optimal offline algorithm introduced above. We can think of competitive analysis as a game in which an *online player* must compete with a malicious *adversary*, also called the *offline player*, who tries to maximize the ratio of online over offline cost. The adversary's task is twofold: he chooses a worst-case instance, and he has to process it in an optimal way.

Given a deterministic online algorithm, we are interested in finding the smallest constant c such that the algorithm is c-competitive.

Definition 2.8 (Competitive Ratio). *Let* ALG *be a deterministic competitive online algorithm for the online optimization problem* Π*. The* competitive ratio *of* ALG *on* Π *is the infimum over all* $c \geq 1$ *such that* ALG *is* c*-competitive.*

In large parts of this thesis, we will omit the word "strictly", although the constant b can indeed be chosen equal to zero. When it is of interest to allow a positive constant b, we will explicitly point this out.

2.2.2 Randomized Algorithms

If an online algorithm uses randomization, we must be more specific about the adversary it is playing against. The power of the adversary depends on both the degree of information he has when choosing the worst-case sequence and the rules he must obey when serving it.

In [22], Ben-David, Borodin, Karp, Tardos, and Wigderson have proposed three different types of adversaries. We only introduce the most common one here, called the *oblivious adversary*. The other two adversary types are defined in Section A.2, where we also review the known results about their relative powers. These results show that the oblivious adversary is the weakest and the most interesting type of adversary.

Definition 2.9 (Oblivious Adversary). *The* oblivious adversary OBL *knows the probability distribution used by the randomized online algorithm, but it must construct its worst-case sequence σ without knowing the outcomes of the random experiments.* OBL *serves the sequence in the end, after having finished its construction, in an optimal manner w.r.t. cost. We denote the cost of* OBL *on input σ by* $\mathsf{OBL}(\sigma)$.

Competitiveness against this adversary is defined as follows.

Definition 2.10 (Competitiveness against the Oblivious Adversary). *Let $c \in \mathbb{R}$, $c \geq 1$. A randomized algorithm* RALG *for an online optimization problem Π is* c-competitive against the oblivious adversary*, if there exists a constant $b \geq 0$ such that*

$$\mathbb{E}[\mathsf{RALG}(\sigma)] \leq c \cdot \mathsf{OBL}(\sigma) + b$$

for all input sequences σ. The expectation is taken over the random choices made by RALG.

In this thesis, we always consider the oblivious adversary when applying competitive analysis to randomized algorithms. Hence, whenever speaking of the competitive ratio of a randomized online algorithm in the sequel, we assume that it is competing with the oblivious adversary.

2.2.3 Yao's Principle

We now introduce a useful technique for proving lower bounds on the competitive ratio of randomized online algorithms. Lower bounds on the competitive ratio of randomized online algorithms are also called *randomized lower bounds*.

To obtain a lower bound on the competitiveness of any online algorithm, we must provide for each online algorithm a worst-case sequence and compute the (expected) cost of the algorithm on that instance. This proves to be much more difficult

for randomized than for deterministic online algorithms. Fortunately, the following variant of Yao's Principle from game theory makes the construction of randomized lower bounds easier.

Suppose that the set of possible request sequences for a given online problem is $\{ \sigma_x \mid x \in \mathcal{X} \}$ for a suitable index set $\mathcal{X}$. Analogously, let $\{ \mathsf{ALG}_y \mid y \in \mathcal{Y} \}$ be the set of deterministic online algorithms for the problem. Furthermore, we define the *expected cost on a randomized sequence* of a deterministic algorithm ALG_y w.r.t. the distribution X on $\mathcal{X}$ by

$$\mathbb{E}_X \left[\mathsf{ALG}_y(\sigma_x) \right] := \int_{\mathcal{X}} \mathsf{ALG}_y(\sigma_x) \, dX(x).$$

With this notation, we can state the version of Yao's Principle applicable to the online setting as follows.

Theorem 2.11 (Yao's Principle). *Let $\bar{c} \in \mathbb{R}$, $\bar{c} \geq 1$ and $\{\sigma_x \mid x \in \mathcal{X}\}$ be the set of input sequences for the considered online optimization problem. If $\bar{X}$ is a distribution over $\mathcal{X}$ such that*

$$\mathbb{E}_{\bar{X}} \left[\mathsf{ALG}_y(\sigma_x) \right] \geq \bar{c} \cdot \mathbb{E}_{\bar{X}} \left[\mathsf{OPT}(\sigma_x) \right]$$

holds for all deterministic algorithms ALG_y, $y \in \mathcal{Y}$, then $\bar{c}$ is a lower bound on the competitive ratio of any randomized online algorithm against the oblivious adversary.

Hence, in order to prove a lower bound, it suffices to provide one *randomized* sequence on which the expected cost of any *deterministic* online algorithm is high in comparison to the optimal offline cost. We will apply this technique in all our constructions for randomized lower bounds.

It goes beyond the scope of this thesis to introduce all game-theoretic notions and theorems needed for an exact proof of Theorem 2.11. In Section A.3, we give a proof sketch and show how to verify that the necessary preconditions are satisfied for Yao's Principle in its game-theoretic form. Readers not familiar with basic game-theory are referred to the book by Vorob'ev, see [105].

2.3 Weaknesses, Modifications, and Extensions of Competitive Analysis

Competitive analysis is of substantial theoretical value, because it provides a measure for what is lost in the worst case due to lack of information. Nevertheless, competitive analysis has rightly been criticized. One of its drawbacks is to ignore computational complexity issues completely. In fact, competitive analysis allows an algorithm to draw on unlimited computational resources. This is an unrealistic assumption when

designing online algorithms intended for practical use. Disregarding restrictions on computational resources is but one shortcoming of competitive analysis. Its major weakness is being overly pessimistic: the adversary is often too powerful to allow for a distinction of online algorithms. It is not uncommon that two online algorithm achieve the same competitive ratio, although one outperforms the other on inputs relevant for practice. Moreover, some problem classes do not even allow for any competitive online algorithm at all, since the adversary can force the online algorithm to have positive cost, whereas he himself can serve the sequence with zero cost. This is, for instance, the case in the Online Dial-a-Ride Problem with the maximum flow time as objective function, see Chapter 5.

Several modifications and extensions of competitive analysis have been proposed as a remedy. *Comparative analysis*, introduced by Koutsoupias and Papadimitriou in [66], removes some of the adversary's power, for instance by restricting its lookahead. In the *diffuse adversary* model, also proposed in [66], the adversary must create an input according to a probability distribution from a class of distributions that is known to the online algorithm. Then, the worst-case ratio of the expected objective value achieved by the online algorithm over that of an optimal offline algorithm is computed. By *resource augmentation*, all alterations of the model are referred to in which the online player's restrictions on the resources are looser than those that must be obeyed by the offline algorithm. E.g., one might grant the online algorithm a certain lookahead, allow its server to move at faster speed than the optimal offline server, or increase its capacity.

Young introduces the concept of *loose competitiveness* in the context of paging, cf. [110]. It is based on the following two ideas. First, the online algorithm does not have to be c-competitive for all given memory sizes k, but only for a large fraction of them. Second, the ratio of online over offline cost is irrelevant when the absolute cost of the online algorithm is very small. Ajtai, Aspnes, Dwork, and Waarts, cf. [4], investigate a refinement of competitive analysis for a distributed environment. In a distributed setting, there are additional sources of non-determinism other than the request sequence. Rather than comparing the cost of a distributed online algorithm to that of an optimal global-control offline algorithm, the authors suggest to compare it to an optimal distributed algorithm that has to pay extra to learn about other parts of the network.

Another possibility to obtain a meaningful competitive ratio is to impose *problem specific restrictions* on the adversary. In this thesis, we extend existing and develop new concepts to restrict the adversary's power in Online Dial-a-Ride Problems (see Chapter 5). It sometimes also pays to restrict the set of possible input sequences in order to rule out pathological cases. The concept of *reasonable load* for Online-Dial-a-Ride Problems, proposed in [54] by Hauptmeier, Krumke, and Rambau, falls into this category. Finally, Becchetti, Leonardi, Marchetti-Spaccamela, Schäfer, and Vredeveld have recently introduced *smoothed competitive analysis*, a step towards an

average-case analysis, cf. [21]. Here, the input instance is chosen by the adversary but afterwards disturbed according to a specified distribution, and the expected ratio of online over offline cost is computed for the resulting random input. The authors use smoothed competitive analysis to evaluate the Multi-Level Feedback algorithm for minimizing the total flow time on a sequence of jobs whose processing times only become known when they are completed.

2.4 Simulation

In the design, enhancement, and optimization of a real system and its processes, it is often desirable to make predictions on how the alteration of one or several components influences the whole system. Alas, it is often too costly or even impossible to conduct the necessary experiments on the real system, and a tractable mathematical model that captures all important aspects is out of reach. In this case, *simulation* is one resort. In the Merriam-Webster online dictionary (see [77]), simulation is defined as *"the imitative representation of the functioning of one system or process by means of the functioning of another"*.

When simulation is performed by means of a computer, only finitely many aspects of the real system can be modeled. It is a non-trivial task to find out which components and processes are decisive, and it is often time-consuming to develop a good simulation model that reflects all the necessary interactions correctly. Yet, simulation can be of substantial benefit for the understanding of a system. Moreover, it is sometimes the only means to study the effect of optimization strategies on the system, and it is very helpful in their design and visualization. Simulation models often allow to obtain results within shorter time periods than possible in reality, since model time usually runs faster than real time. They are flexible in the sense that one can alter the model easily, for instance, such as to take stochastic aspects into account.

All these arguments provide strong support for the use of simulation when optimizing real systems. One must be aware, however, that simulation does not provide worst-case, but only experimental performance guarantees. Thus, conclusions derived from simulation depend heavily on the experimental set-up. It is particularly important to use real data in simulation experiments whenever possible.

Simulation models can be classified into several types. In this thesis, we are only concerned with *discrete event simulation*, which is used to describe systems where changes occur at distinguished points in time. Ascheuer discusses different types of simulation and gives a detailed description how to build a discrete event simulation model in [8]. It is supplemented by the software package AMSEL (cf. [7]), a library for discrete event simulation. An introduction to discrete event simulation can also be found in the book [91] by Siegert.

Part I

New Results in Online Dial-a-Ride Problems