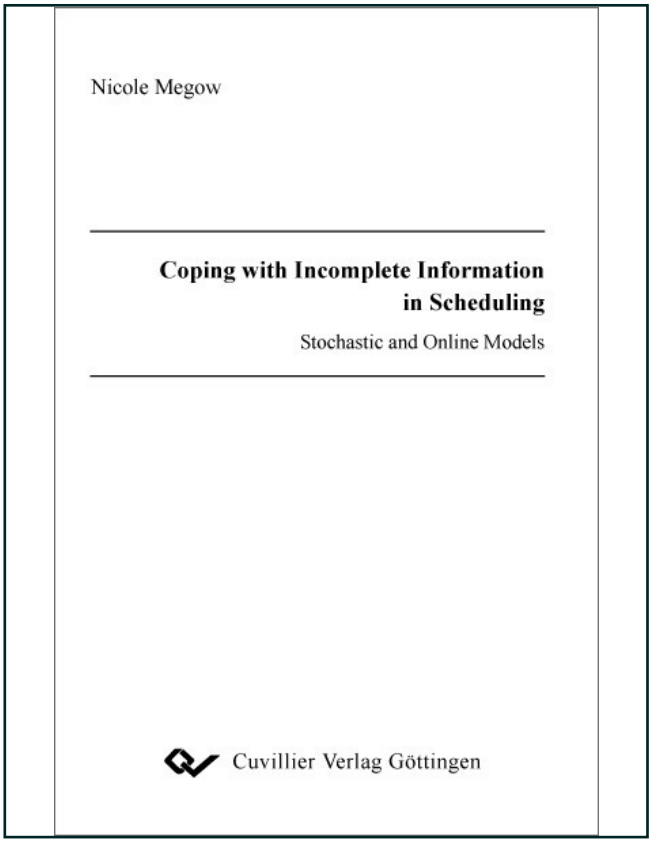
Coping with Incomplete Information in Scheduling — Stochastic and Online Models



Introduction

Incomplete information is an omnipresent issue when dealing with real-world optimization problems. Typically, such limitations concern the uncertainty of given data or the complete lack of knowledge about future parts of a problem instance. This thesis is devoted to investigations on how to cope with incomplete information when solving scheduling problems.

To cope with scenarios where there is uncertainty about the problem data or parts of the instance are not known at all, there are two major frameworks in the theory of optimization. One deals with *stochastic information*, the other with *online information*. In the stochastic information environment, probabilistic assumptions are made on the data set or parts of it. Usually, the distribution functions of the respective random variables are assumed to be known in advance before executing an optimization algorithm. The consideration of stochastic data is a major step in terms of bridging the gap from theory to practice. However, the decisive assumption on the a priori knowledge of the entire instance and the distributions appears quite strong for certain applications. In contrast, in an online information environment, the assumption is that a problem instance is presented to an algorithm only piecewise. The actual problem data are usually disclosed upon arrival of a request, and decisions must be made based on partial knowledge of the instance. This second model appears more restrictive in the informational sense since it models the complete lack of knowledge about the future. On the other hand, once the data are revealed, they are usually completely certain.

Both frameworks have their legitimacy depending on the actual application. Nevertheless, problem settings are conceivable that comprise both, uncertain information about the data set and the complete lack of knowledge about the future. This rouses the need for a generalized model that integrates both traditional information environments.

In this thesis, we deal with all three of the mentioned frameworks, the classic ones with stochastic and online information, respectively, as well as a generalized model that combines both traditional models in a joint framework.

The particular problem class we consider is the class of scheduling problems which plays an important role within combinatorial optimization. These problems involve the temporal allocation of limited resources for executing

activities so as to optimize some objective. Scheduling problems are apparent in many applications including, for example, manufacturing and service industries but also compiler optimization and parallel computing.

Practitioners and researchers are naturally interested in objective functions that are related to the minimization of completion times. The two classic types to mention here are the makespan, which is the completion time of the final job in the system, and the total (weighted) completion time of all jobs. Clearly, both types provide satisfying information on the scheduling performance for different applications. However, the individual job related objective function – the sum of all completion times – captures important details on the operations efficiency which are not reflected in the makespan objective. Therefore, we focus in this thesis mainly on problems to minimize the sum of weighted completion times. Only in the last chapter we deviate from this procedure. We not only consider a deadline based objective, we also leave the standard scheduling framework and add a locality component; that means, scarce resource must be allocated over time and, in addition, their position in a metric space must be respected.

These problems are $\mathcal{NP}$-hard, in general. Thus, by the current state-of-the-art, we cannot hope to find efficient algorithms that solve these problems to optimality. Therefore, we are restricted to find near-optimal solutions efficiently. A widely accepted measure for the performance of an algorithm is the approximation ratio, that is, the ratio of the outcome of an algorithm over the optimum value. In the past three decades, approximation algorithms for many $\mathcal{NP}$-hard optimization problems were designed and analyzed. Interestingly, one of the first approximation algorithms with a formalized analysis is due to Graham [Gra69] and was designed for a machine scheduling problem.

Certainly, these problems do not lose their intractability when information about the problem data is limited. Performance measures comparable to the approximation ratio, exist for both traditional restricted information environments, the online and the stochastic model – even though there is less agreement on the "right" measure. Within these frameworks, we design algorithms for certain scheduling problems. Thereby we provide first constant performance guarantees or improve previously best known results.

A general model that combines stochastic and online information is certainly of interest and can be designed as a natural extension. But the challenging question is whether there exists any algorithm that can perform well in such a restricted information environment. More precisely, is there an algorithm that yields a constant performance guarantee? We successfully treat this intriguing question and give a positive answer by providing such algorithms for machine scheduling problems. In fact, our results are compet-

itive with the performance guarantees best known in the traditional settings of stochastic and online scheduling. Thus, they do not only justify the generalized model but also imply – at least in the considered problem settings – that optimization in the general model with incomplete information does not necessarily mean to give up performance.

Outline of the Thesis

Chapter 1. In this preliminary chapter we introduce basic concepts for modeling incomplete information on problem data. For both standard models that we deal with, the online and the stochastic model, we give details on terminology and typical (as well as alternative) performance measures. Furthermore, we introduce the problem classes that we focus on in this thesis; this includes the diverse class of *scheduling problems* as well as a brief overview on classic *traveling salesman problems*.

Chapter 2. The second chapter concerns online scheduling problems. We investigate algorithms for scheduling with the objective to minimize the total weighted completion time on single as well as on parallel machines. We consider both, a setting with independent jobs and one where jobs must obey precedence relations.

For independent jobs arriving online, we design and analyze algorithms for both, the preemptive and the non-preemptive setting. These online algorithms are extensions of the classical Smith rule (Wspt algorithm) and yield performance guarantees that are improving on the previously best known ones. A natural extension of Smith's rule to the preemptive setting is 2-competitive. Nevertheless, we propose a modified, more dynamic variant of this classic algorithm for the single-machine problem which achieves the same competitive ratio. While the performance analysis of the first algorithm is proven to be tight, the lower bound of our dynamic rule leaves a gap that raises hope that its true performance is much better. For the non-preemptive variant of the multiple-machine scheduling problem, we derive a 3.281-competitive algorithm that combines a processing time dependent waiting strategy with Smith's rule.

For the scenario in which precedence constraints among jobs are given, we discuss a reasonable online model and give lower and upper bounds on the competitive ratio for scheduling without job preemptions. In this context, previous work on the offline problem of scheduling jobs with generalized precedence constraints, the so called And/Or-precedence relations, appears to be adoptable to a certain extent; we discuss the relevance.

CHAPTER 3. The third chapter is devoted to the stochastic scheduling model. After reviewing (approximation) results in this area, we focus on preemptive stochastic scheduling. We present a first constant approximation for a preemptive problem to minimize the sum of weighted completion times. For scheduling jobs with release dates on identical parallel machines we derive a policy with a guaranteed performance ratio of 2 which matches the currently best known result for the corresponding deterministic online problem (derived in the previous chapter).

In contrast to the previously known results in the non-preemptive setting, our preemptive policy extensively utilizes information on processing time distributions other than the first (and second) moments. In order to derive our result we introduce a new non-trivial lower bound on the expected value of an unknown optimal policy. This bound is derived by borrowing ideas for a *fast single-machine relaxation* known from deterministic online scheduling. The crucial ingredient to our result is then the application of a Gittins index priority policy which is optimal to a relaxed version of our fast single-machine relaxation. This priority index also inspires the design of our policies.

CHAPTER 4. In this chapter, we consider a model for scheduling with incomplete information which combines the main characteristics of online and stochastic scheduling, as considered in the previous two chapters, in a simple and natural way. Job processing times are assumed to be stochastic, but in contrast to the traditional stochastic scheduling model, we assume that jobs arrive online, and there is no knowledge about the jobs that will arrive in the future. We name this model the *stochastic online scheduling model*. Indeed, it incorporates both, stochastic scheduling and online scheduling as a special case – also with respect to the performance measure.

The particular problems we consider are preemptive and non-preemptive parallel-machine scheduling with the objective to minimize the total weighted completion times of jobs. For the problem where jobs must run until completion without interruption, we analyze simple, combinatorial online scheduling policies and derive performance guarantees that match the currently best known performance guarantees for stochastic and online parallel-machine scheduling. For processing times that follow NBUE distributions, we improve upon previously best known performance bounds from stochastic scheduling, even though we consider a more general setting.

Finally, we argue that the results on preemptive stochastic (offline) scheduling presented in the previous chapter also apply in this more general model because the preemptive policy is feasible in an online setting as well.

Chapter 5. In the final chapter, we turn back to the traditional online optimization model. However, we deviate slightly from our pure scheduling focus and consider a variant of an online *traveling salesman problem* (Tsp). In this problem, requests with deadlines arrive online over time at points of a metric space. One or more servers move in a metric space at constant speed and serve requests by reaching the corresponding position before the deadline. The goal is to guide the server through the metric space such that the number of served requests is maximized. Thus, we have a Tsp-like problem with a maximization objective.

Applying standard competitive analysis, we study the problem on restricted metric spaces, namely the real line and the uniform metric space. While on the line no deterministic algorithm can achieve a constant competitive ratio, we provide competitive algorithms for the uniform metric space. Our online investigations are complemented by complexity results for the offline problem.

Even though this problem is not a standard scheduling problem, we associate it with this problem class in a broader sense. In a way, Tsp-like problems can be interpreted as scheduling problems with an additional locality component: an algorithm decides about the timing of the server moves, and in addition, it determines the direction of moves. However, the locality aspect becomes less important on the restricted spaces that we consider.

Chapter 1

Preliminaries

We investigate optimization problems in different information environments. They all have in common that a solution must be found based on incomplete information on the problem input. The classical ways to model such limitations of knowledge is to assume *online* or *stochastic* information.

In this chapter, we give details on both standard information environments and introduce the optimization problems that we consider in this thesis. After separate, detailed examinations on scheduling problems in both, an online and stochastic setting, in the following two chapters, we will discuss a combined model of *stochastic online information* and successfully analyze scheduling problems within this model in Chapter 4.

1.1 Introduction of Problems

In the major part of this thesis, we focus on a certain class of scheduling problems. In the following Section 1.1.1 we give an overview including a classification scheme and some definitions. In Section 1.1.2 we introduce a second class of well-known optimization problems, namely the traveling salesman problem with different objective functions.

1.1.1 Scheduling Problems and Standard Notation

Scheduling problems have been studied extensively for decades and are still of ongoing interest. The universal problem of assigning tasks temporally to scarce resources occurs in endless variations in the real world. Therefore, a remarkable amount of different scheduling models has been introduced in the literature. We refer to a recent and quite exhaustive collection of surveys on various models and problem settings edited by Leung [Leu04].

In this thesis, we consider certain problems within the class of so-called *machine scheduling problems*. In general, such problems can be described as follows: There is given a set $\mathcal{J} = \{1, 2, \ldots, n\}$ of jobs which must be scheduled on one or more machines. Each job $j \in \mathcal{J}$ has associated a positive

processing time $p_j \geq 0$ and a non-negative weight $w_j \geq 0$. The goal is to find an assignment of jobs to time slots and machines – which we call a *schedule* – such that certain problem specific constraints are met and an objective function is minimized.

Typically, (machine) scheduling problems are described using a standard classification scheme that was initially introduced by Graham, Lawler, Lenstra, and Rinnooy Kan [GLLR79], which is also called the three-field notation $\alpha \,|\, \beta \,|\, \gamma$. For the sake of completeness we review this standard notation but focus on scheduling problems that we consider in this thesis.

The parameter α in the first field describes the machine environment. We consider the following two choices:

- 1 *(single machine).* Jobs must be scheduled on a single processor which can process only one job at a time.
- P *(parallel identical machines).* Each job can be processed on any of the identical parallel machines. The number of available machines is denoted by m. Each of the m machines can handle only one job at a time.

The β field represents job characteristics that are given to each job j in addition to its processing time p_j and weight w_j.

- r_j *(release date).* Each job has associated a release date; without this parameter, we assume $r_j = 0$ for all jobs j. The release date defines the earliest point in time when a job is available for processing. In our online models, this will also be the time when the scheduler learns about the existence of the job.

  If not stated differently, we assume for convenience that jobs are indexed such that $r_j \leq r_k$ for $j < k$.
- *pmtn (preemption).* Including this parameter indicates that job preemption is allowed. That means that the processing of a job may be suspended and resumed later on any machine at no extra cost. In contrast, if preemption is not permitted (that is, if the parameter is not set) then a job that has started must run until its completion on the same machine without any interruption.
- *prec (precedence constraints).* Precedence constraints define a partial order $(\mathcal{J}, \prec)$ on the set of jobs $\mathcal{J}$. Jobs have to be scheduled in compliance with these constraints, where $j \prec k$ implies that job k must not start processing before j has completed.

If no precedence constraints are given, then we call the jobs independent.

– d_j *(deadline).* The deadline of a job represents the date by which a job must be completed. If a job has not finished until its deadline, then it is called *late*. If this parameter is not included in the β-field, then all deadlines are assumed to be infinite, that means, they are non-existent.[1]

Finally, the third field, γ, indicates the objective function. We mainly consider one objective function in different models for scheduling with limited information.

– $\sum w_j C_j$ (*minimize the sum of weighted completion times*). Here, C_j denotes the completion time of a job j. If all job weights are equal, then this objective is equivalent to minimizing the average completion time.

– $\mathbb{E}\left[\sum w_j C_j\right]$ (*minimize the sum of weighted completion times in expectation*). In the stochastic scheduling environment (in Chapters 3 and 4) our goal is to minimize the objective function above in expectation. By having an expected objective value in the γ field we also denote that the jobs have stochastic processing times which is in some literature denoted by $p_j \sim \mathit{stoch}$ in the β field. Since the only stochastic problem characteristics in this work are stochastic processing times, we omit this truly exact – but in our case redundant – description.

Clearly, the relevance of certain objective functions is discussible and depends on the particular application. An objective that has gained a fair amount of attention recently is to *minimize the sum of (weighted) flow times*, $\sum(w_j) F_j$, where the flow time is defined as $F_j = C_j - r_j$. Problems of this type seem best to reflect the goal of certain scheduling problems; on the other hand, they are extremely intractable. The probably most widely studied objective is to minimize the latest completion time among all jobs, $C_{\max} := \max_{j \in \mathcal{J}} C_j$, also called the *makespan* of a schedule. Finally, we mention the objective to *minimize the number of late jobs*, $\sum U_j$, which turns out to be relevant for a discussion in Chapter 5; here, $U_j = 1$ indicates that the job j is late, that is, $C_j > d_j$, and otherwise we have $U_j = 0$. Obviously, this objective coincides with aiming for maximizing the number of jobs that are completed on time, that is, when $C_j \leq d_j$. (While in the minimization problem all jobs must

[1] We do not explicitely investigate scheduling problems with deadlines. However, we will discuss the close relation between such problems and the Tsp-related problem considered in Chapter 5.