Coping with Incomplete Information in Scheduling – Stochastic and Online Models*

Nicole Megow

Institut für Mathematik, Technische Universität Berlin, Germany
nicole.megow@tu-berlin.de

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. Our work is devoted to investigations on how to cope with incomplete information when solving scheduling problems. The particular problem class we consider is the class of machine scheduling problems which plays an important role within combinatorial optimization. These problems involve the temporal allocation of limited resources (machines) 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.

There are two major frameworks for modeling limited information in the theory of optimization. One deals with *stochastic information*, the other with *online information*. Within these models, we design algorithms for certain scheduling problems. Thereby we provide first constant performance guarantees or improve previously best known results.

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. Such a general model is designed as a natural extension that combines stochastic and online information. The challenging question is whether there exists any algorithm that can perform well in such a restricted

---

* This work has been supported by the DFG Research Center MATHEON in Berlin.
The book version of this extended abstract is published as [6].

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 certain machine scheduling problems. In fact, our results are competitive 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.

1 Stochastic Scheduling

In stochastic scheduling we assume uncertainty about job processing times. Any job j must be processed for P_j units of time, where P_j is a random variable. We assume that all random variables of processing times are stochastically independent. This restriction on the probability functions is not part of the stochastic scheduling model; still, the independence of random variables is crucial for our and previously known results.

The solution of a stochastic scheduling problem is not a simple schedule, but a so-called *scheduling policy*; see [10]. A policy must not anticipate information about the future, such as the actual realizations of the processing times of the jobs that have not yet been completed; we say a stochastic scheduling policy must be *non-anticipatory*.

Various research on stochastic scheduling has been published concerning criteria that guarantee the optimality of simple policies for rather special, restricted scheduling problems. Only recently research interest addressed also approximative policies [11, 13, 2]. While all of the results hold for non-preemptive scheduling, we are not aware of any approximation results for problems that allow job preemption except from the optimality of the Gittins index priority policy [12, 15, 4] for the problem $1 \,|\, \mathrm{pmtn} \,|\, \mathbb{E}\left[\sum w_j C_j\right]$.

We derive first constant approximation guarantees for preemptive stochastic scheduling policies on multiple machines and/or individual release dates. For jobs with general processing time distributions, we give a 2-approximative policy for minimizing the expected sum of weighted completion times.

In order to derive our results we introduce a new non-trivial lower bound on the expected value of an unknown optimal policy. This bound is obtained borrowing ideas for a *fast single-machine relaxation* [1]. The crucial ingredient to our investigations is then the application of the above mentioned Gittins index priority policy which solves a relaxed version of our fast single-machine relaxation optimally [12, 15, 4]. The priority index used in this policy also inspires the design of our policies. Thereby, our preemptive policies extensively utilize information on processing time distributions other than the first (and second) moments, which distinguishes them significantly from approximative policies known in the non-preemptive setting.

The Gittins index is defined as follows. Given that a job j has been processed for y time units and it has not completed, we define the *expected investment* of processing this job for q time units or up to completion, which ever comes first, as

$$I_j(q,y) = \mathbb{E}\,[\,\min\{\mathrm{P}_j - y, q\}\,|\,\mathrm{P}_j > y\,]\,.$$

The ratio of the weighted probability that this job is completed within the next q time units over the expected investment, is the basis of the Gittins index priority rule. We define it as the *rank* of a sub-job of length q of job j, after it has completed y units of processing:

$$R_j(q,y) = \frac{w_j \mathrm{Pr}\,[\mathrm{P}_j - y \leq q\,|\,\mathrm{P}_j > y]}{I_j(q,y)}.$$

This ratio is well defined if we assume that we compute the rank only for $q > 0$ and $\mathrm{P}_j > y$, in which case the investment $I_j(q,y)$ has a value greater than zero.

For a given (unfinished) job j and attained processing time y, we are interested in the maximal rank it can achieve. We call this the Gittins index, or rank, of job j, after it has been processed for y time units.

$$R_j(y) = \max_{q \in \mathbb{R}^+} R_j(q,y).$$

With the definitions above, we define a policy based on the rank for scheduling on parallel machines where jobs have release dates.

Follow Gittins Index Priority Policy (F-Gipp): At any time t, process an available job j with highest rank $R_j(y_{j,k+1})$, where (j,k) is the last quantum of j that has completed and $y_{j,k+1}$ is the amount of processing that has been completed before the next quantum $(j,k+1)$ starts. Define $k = 0$ if no quantum of job j has been completed.

The policy F-Gipp is a 2-approximation for the preemptive stochastic scheduling problem $\mathrm{P}\,|\,r_j, pmtn\,|\,\mathbb{E}\,[\sum w_j C_j\,]$. However, on restricted problem instances it coincides with policies whose optimality is known; see [9, 6].

2 Online Scheduling

In online scheduling we assume that jobs and their characterizing data become known to the scheduler only piecewise. Thus, an online algorithm must take scheduling decisions based only on the partial knowledge of the instance as it is given so far.

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 [14] 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. 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.

We are not aware of any existing results 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 [3], appears to be adoptable to a certain extent.

3 Stochastic Online Scheduling

We consider the *stochastic online scheduling* (Sos) model that generalizes both traditional models for dealing with incomplete information, stochastic scheduling and online scheduling. Like in online scheduling, we assume that the instance is presented to the scheduler piecewise, and nothing is known about jobs that might arrive in the future. Even the number of jobs is not known in advance. Once a job arrives, we assume, like in stochastic scheduling, that the probability distribution of its processing time is disclosed, but the actual processing time remains unknown until the job completes.

The goal is to find an Sos policy that minimizes the expected objective value. Our definition of a stochastic online scheduling policy integrates the traditional definition of stochastic scheduling policies into the setting where jobs arrive online. In order to decide, such a policy may utilize the complete information contained in the partial schedule up to time t. But it must not utilize any information about jobs that will be released in the future and it must not use the actual processing times of scheduled (or unscheduled) jobs that have not yet completed. In the performance evaluation we also generalize the definitions of an approximative policy for stochastic scheduling and a competitive algorithm in online scheduling; see [8, 6]. In this view, our model somewhat compares to the idea of a *diffuse adversary* as defined by Koutsoupias and Papadimitriou [5].

Various (scheduling) problems can be modeled in this stochastic online setting. We consider the particular settings of preemptive and non-preemptive scheduling with the objective to minimize the expected total weighted completion times of jobs.

For the problem where jobs must run until completion without interruption, $\mathrm{P}\,|\,r_j\,|\,\mathbb{E}\,[\sum w_j C_j\,]$, 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, a MinIncrease policy even improves upon previously best known performance bounds from stochastic scheduling, even though it is feasible in a more general setting. This policy assigns each job j to the machine where it causes the least increase in the expected objective value, given the previously assigned jobs (when release dates are ignored). In the analysis we exploit the fact that the lower bound for an optimal policy in the traditional stochastic scheduling environment in [11] is by definition also a lower bound for an optimal policy in the Sos model.

In the preemptive setting we can argue that the 2-approximative policy for preemptive stochastic (offline) scheduling in Section 1 for $\mathrm{P}\,|\,r_j, \mathrm{pmtn}\,|\,\mathbb{E}\,[\sum w_j C_j\,]$ also applies in this more general model because the preemptive policy is feasible in an online setting as well. Moreover, the currently best known online algorithm for deterministic processing time has also a competitive ratio of 2; see [7].

Conclusion.

These results do not only justify the general model for scheduling with incomplete information. They also show for certain scheduling problems that policies designed to deal with stochastic *and* online information, can achieve the same theoretic performance guarantee as policies that can handle only one type of limited knowledge.

References

1. C. Chekuri, R. Motwani, B. Natarajan, and C. Stein. Approximation techniques for average completion time scheduling. *SIAM Journal on Computing*, 31:146–166, 2001.
2. B. C. Dean. *Approximation Algorithms for Stochastic Scheduling Problems*. PhD thesis, Massachusetts Institute of Technology, 2005.
3. T. Erlebach, V. Kääb, and R. H. Möhring. Scheduling AND/OR-networks on identical parallel machines. In *Proc. of the First International Workshop on Approximation and Online Algorithms, WAOA 2003*, volume 2909 of *Lecture Notes in Computer Science*, pages 123–136, 2004.
4. J. C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society*, Series B, 41:148–177, 1979.
5. E. Koutsoupias and C. H. Papadimitriou. Beyond competitive analysis. *SIAM Journal on Computing*, 30:300–317, 2000.
6. N. Megow. *Coping with Incomplete Information in Scheduling – Stochastic and Online Models*. Dissertation 2006, Technische Universität Berlin. Cuvillier Göttingen, 2007.
7. N. Megow and A. S. Schulz. On-line scheduling to minimize average completion time revisited. *Operations Research Letters*, 32(5):485–490, 2004.
8. N. Megow, M. Uetz, and T. Vredeveld. Models and algorithms for stochastic online scheduling. *Mathematics of Operations Research*, 31(3):513–525, 2006.
9. N. Megow and T. Vredeveld. Approximation in preemptive stochastic online scheduling. In *Proc. of 14th European Symposium on Algorithms*, number 4168 in Lecture Notes in Computer Science, pages 516–527, 2006.
10. R. H. Möhring, F. J. Radermacher, and G. Weiss. Stochastic scheduling problems I: General strategies. *Zeitschrift für Operations Research*, 28:193–260, 1984.
11. R. H. Möhring, A. S. Schulz, and M. Uetz. Approximation in stochastic scheduling: the power of LP-based priority policies. *Journal of the ACM*, 46:924–942, 1999.
12. K. C. Sevcik. Scheduling for minimum total loss using service time distributions. *Journal of the ACM*, 21:65–75, 1974.
13. M. Skutella and M. Uetz. Stochastic machine scheduling with precedence constraints. *SIAM Journal on Computing*, 34(4):788–802, 2005.
14. W. E. Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3:59–66, 1956.
15. G. Weiss. On almost optimal priority rules for preemptive scheduling of stochastic jobs on parallel machines. *Advances in Applied Probability*, 27:827–845, 1995.