Stochastic Online Scheduling Revisited

Andreas S. Schulz

Sloan School of Management, Massachusetts Institute of Technology,
E53-361, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

Abstract. We consider the problem of minimizing the total weighted completion time on identical parallel machines when jobs have stochastic processing times and may arrive over time. We give randomized as well as deterministic online and off-line algorithms that have the best known performance guarantees in either setting, deterministic and off-line or randomized and online. Our analysis is based on a novel linear programming relaxation for stochastic scheduling problems, which can be solved online.

1 Introduction

We study approximation algorithms for stochastic and online versions of the following deterministic, off-line scheduling problem. There is a set of n jobs to be processed on m identical parallel machines. Each job j has a nonnegative weight w_j, processing time p_j, and release date r_j. After its release, a job has to be processed on some machine, and each machine can handle at most one job at a time. The objective is to assign jobs to machines and to determine a feasible sequence on each machine so as to minimize the total weighted completion time, $\sum_{j=1}^{n} w_j C_j$. Here, C_j denotes the completion time of job j in the schedule. The deterministic problem is well understood: It is known to be strongly NP-hard (Lenstra, Rinnooy Kan, and Brucker 1977), and it has a polynomial-time approximation scheme (Afrati et al. 1999); a simpler 2-approximation algorithm, which is of particular relevance to the work described here, was earlier given by Schulz and Skutella (20002b).

In stochastic scheduling (Möhring, Radermacher, and Weiss 1984), job processing times are modeled as random variables, each specified by some probability distribution (with expected value μ_j and standard deviation σ_j). The actual processing time of a job does not become known before it is completed. Research has traditionally focused on nonanticipative policies that aim at minimizing the objective function in expectation. Moreover, it is typically assumed that job processing times are stochastically independent. These views are adopted here as well. A scheduling policy is nonanticipative if its decisions about which jobs to schedule at any given time t depend only on the jobs that are already completed by that time and on the conditional distributions of the remaining processing times of jobs that are still active at time t.

For the single-machine problem without nontrivial release dates ($m = 1, r_j = 0$ for all jobs j), Rothkopf (1966) showed that the WSEPT rule is optimal, which schedules the jobs in order of nonincreasing ratios of weight to expected processing time. For unit weights and exponentially distributed processing times, the Shortest Expected Processing Time rule remains optimal on identical parallel machines (Weiss and Pinedo 1980). In fact, Weber, Varaiya, and Walrand (1986) showed that it suffices when the processing time distributions are stochastically comparable in pairs. For arbitrary weights, WSEPT is optimal for exponentially distributed processing times if the WSEPT order of jobs coincides with sequencing the jobs in the order of nonincreasing weights (Kämpke 1987). Under minor technical assumptions, Weiss (1990) showed that the WSEPT rule is asymptotically optimal.

The stochastic scheduling problem considered here, when jobs may have individual release dates, was first addressed by Möhring, Schulz, and Uetz (1999). For processing time distributions whose coefficients of variation σ_j/μ_j are bounded from above by $\sqrt{\Delta}$, they gave a static priority policy whose expected objective function value is within a factor of $\max\{4, 3+\Delta\}$ of that of an optimal policy.[1] In addition, they showed that the WSEPT rule has a performance guarantee of $1 + (\Delta + 1)/2$ for the problem with identical release dates. This marked the first time that the use of approximation algorithms was proposed in the realm of stochastic scheduling. The analysis as well as the algorithm for the general case is based on a linear programming relaxation, which provides a lower bound on the expected value of an optimal policy.

A different way of dealing with incomplete information is that of online algorithms and competitive analyses. In our context, jobs arrive over time and are completely unknown prior to their arrival. However, a job's processing time and weight are fully revealed at the time of its arrival. The performance of an online algorithm is usually compared to that of an optimal off-line algorithm, which has full hindsight. This quotient is known as the competitive ratio. For randomized online algorithms, we compare the expected objective function value of the solution generated by the algorithm to the value of an off-line optimum. This corresponds to the so-called oblivious adversary model. We refer the reader to Borodin and El-Yaniv (1998) for a general introduction to online algorithms, and to Sgall (1998) for a survey of online scheduling models and results. In the context of the identical parallel machine scheduling problem considered here, online algorithms were designed and analyzed by Hall et al. (1997), Chakrabarti et al. (1996), Schulz and Skutella (20002b), Megow and Schulz (2004), and Correa and Wagner (2008). The currently best deterministic online algorithm has a competitive ratio of 2.618 (Correa and Wagner 2008), while the best known randomized algorithm is 2-competitive (Schulz and Skutella 20002b).

Chou et al. (2006) proposed to study stochastic online scheduling, where jobs arrive over time, as in online scheduling, but when a job arrives only its weight and processing time distribution become known. The expected total weighted completion time of the schedule computed by an online policy is then compared to that of an optimal stochastic policy, which has access to all job release dates,

[1] The performance guarantee of this algorithm is actually slightly better than this; to make for an improved reading, we generally suppress terms of order $1/m$ from this extended abstract.

weights, and processing time distributions at time 0. In other words, the adversary controls the arrival of jobs, their weights, and their processing time distributions, but he cannot influence the actual realization of processing times. While Chou et al. (2006) considered single-machine and flow-shop problems, Megow, Uetz, and Vredeveld (2006) looked at the identical parallel machine model considered here. They introduced δ-NBUE distributions[2] and gave a deterministic online algorithm with performance guarantee $3/2 + \delta + \sqrt{4\delta^2 + 1}/2$. Their analysis uses the linear programming relaxation introduced by Möhring, Schulz, and Uetz (1999), when their algorithm does not.

In this paper, we present deterministic and randomized approximation algorithms that have the best known performance guarantees for stochastic (off-line and online) scheduling on identical parallel machines with the sum of weighted completion times objective. The key is a new, stronger linear programming relaxation for this problem. Moreover, this linear program can be solved by a simple online rule, which constructs a preemptive single-machine schedule. We use this schedule to define an online policy for the original, stochastic problem. This approach has previously been used successfully for various deterministic online problems, including nonpreemptive scheduling on a single machine (Goemans et al. 2002), preemptive single-machine scheduling (Schulz and Skutella 20002a), identical parallel machine scheduling (Schulz and Skutella 20002b), and uniform parallel machine scheduling (Chou, Queyranne, and Simchi-Levi 2006).

We present one randomized and one deterministic algorithm; both work online and run in polynomial time.[3] Their respective performance ratios are $2 + \Delta$ and $\max\{2.618, 2.309 + 1.309\Delta\}$, respectively. The randomized algorithm can be derandomized, which results in a deterministic $(2 + \Delta)$-approximation algorithm for the stochastic (off-line) scheduling problem. Table 1 compares the new results from this paper to earlier results.

The algorithms proposed here are derived from earlier algorithms for deterministic scheduling problems, as were previous algorithms for stochastic scheduling. In our case, we manipulate a randomized online algorithm of Schulz and Skutella (20002b) as well as a deterministic online algorithm by Correa and Wagner (2008). Previously, Möhring, Schulz, and Uetz (1999) built on deterministic algorithms by Hall et al. (1997); Skutella and Uetz (2005) used techniques of Chekuri et al. (2001); and Megow, Uetz, and Vredeveld (2006) drew on ideas

[2] For $\delta = 1$, one recaptures the well-known NBUE distributions, "New Better than Used in Expectation," which include, among others, exponential, Erlang, uniform, and Weibull distributions. In the context of stochastic scheduling, an NBUE distribution would imply that the expected remaining processing time of a job in process is never more than the expected processing time of that job before it was started. NBUE distributions satisfy $\Delta \leq 1$ (Hall and Wellner 1981). In general, $\Delta \leq 2\delta - 1$ (Megow, Uetz, and Vredeveld 2006).

[3] A general definition of the input size of a stochastic scheduling problem would need to deal with the way in which arbitrary probability distributions are specified. The running times of the algorithms proposed here depend only on the input size of the corresponding deterministic problems where job processing times represent expected values.

Table 1. Overview of the development of performance guarantees/competitive ratios for stochastic scheduling with the total weighted completion time objective. To allow for a comparison, we assume that the processing time of each job follows an NBUE distribution.

Model	Performance Guarantee		Reference
	deterministic	randomized	
off-line, all $r_j = 0$	2	–	Möhring et al. (1999)
off-line, general r_j	4	–	Möhring et al. (1999)
	3.618	–	Megow et al. (2006)
	3	–	this paper
online	3.618		Megow et al. (2006)
	3.618	3	this paper

from Megow and Schulz (2004). In each case, the challenge is to refine the algorithm and its analysis such that they still work, even though job processing times are random. In contrast to the previous approximation and online algorithms for stochastic scheduling problems, which all relied on the lower bounds introduced by Möhring, Schulz, and Uetz (1999), we use a linear programming relaxation that is new in the context of stochastic scheduling.

2 A Linear Programming Relaxation

Möhring, Schulz, and Uetz (1999) showed that the vector of expected completion times of any nonanticipative policy satisfies the following inequalities:

$$\sum_{j\in S}\mu_j\, C_j \geq \frac{1}{2m}\Big(\sum_{j\in S}\mu_j\Big)^2 - \frac{\Delta-1}{2}\sum_{j\in S}\mu_j^2 \quad \text{for all } S\subseteq N.$$

As mentioned before, Δ is an upper bound on the squared coefficients of variation; i.e., $\sigma_j^2/\mu_j^2 \leq \Delta$ for all $j\in N$, where $N := \{1,2,\dots,n\}$ denotes the set of all jobs. One can strengthen these inequalities by observing that none of the jobs in a subset S can be processed before time $r_{\min}(S) := \min\{r_j : j\in S\}$.

Lemma 1. *Let Π be a nonanticipative policy for the stochastic identical parallel machine scheduling problem. Then, the corresponding vector $E[C^\Pi]$ of expected completion times satisfies the following inequalities:*

$$\sum_{j\in S}\mu_j\Big(C_j+\frac{\Delta-1}{2}\mu_j\Big) \geq r_{\min}(S)\sum_{j\in S}\mu_j + \frac{1}{2m}\Big(\sum_{j\in S}\mu_j\Big)^2 \quad \textit{for all } S\subseteq N. \tag{1}$$

A similar observation was made earlier in the context of deterministic scheduling; see Queyranne and Schulz (1995). Its relevance in our situation is a consequence of the fact that the associated linear programming relaxation, when we minimize $\sum_{j\in N} w_j C_j$ over (1), is equivalent to that of a deterministic single-machine problem. In fact, setting $M_j := C_j + \frac{\Delta-1}{2}\mu_j$, leads to the following, equivalent linear program:

$$\min \quad \sum_{j\in N} w_j M_j \tag{2a}$$

$$\text{s.t.} \quad \sum_{j\in S} \frac{\mu_j}{m} M_j \geq \frac{\sum_{j\in S}\mu_j}{m}\left(r_{\min}(S) + \frac{\sum_{j\in S}\mu_j}{2m}\right) \qquad \text{for all } S \subseteq N. \tag{2b}$$

Note that in (2) we have dropped the term $\frac{1-\Delta}{2}\sum_{j\in N} w_j\mu_j$ from the objective function, as it is constant anyway.

The linear program (2) can be interpreted as a relaxation of a single-machine scheduling problem where jobs have (deterministic) processing times μ_j/m, and the formulation makes use of mean busy time variables M_j. The mean busy time M_j of job j is the average point in time at which the (single) machine is busy with processing job j. In other words, if $I_j(t)$ is 1 if the machine is processing job j at time t, and 0 otherwise, then

$$M_j = \frac{1}{p_j}\int_{r_j}^{\infty} I_j(t)\, t\, dt \ .$$

Here and henceforth, we use p_j to denote the processing time of job j on the "fast" single machine; i.e., $p_j = \mu_j/m$.

Theorem 2 (Goemans et al. 2002). *The mean busy time vector of the preemptive single-machine schedule that is constructed by the following online algorithm is an optimal solution to the linear programming relaxation* (2)*:*

> *At any point in time, schedule from the available (and as of yet not completed) jobs one with the highest ratio of weight to processing time.*

As was done before (Goemans et al. 2002; Schulz and Skutella 20002a; Chou et al. 2006; Correa and Wagner 2008), we refer to this preemptive schedule as the "LP schedule." It is worth pointing out that Theorem 2 effectively implies that one can solve the linear programming relaxation of minimizing $\sum_{j\in N} w_j C_j$ over (1) online. So, not only it provides a lower bound on the expected value of an optimal off-line policy, but also it can be used to design an online algorithm for the stochastic scheduling problem itself.

3 A Randomized Algorithm

In the spirit of all previous approximation algorithms for nonpreemptive stochastic scheduling problems, which are based on existing algorithms for deterministic scheduling problems, we will now extend an algorithm of Schulz and Skutella (20002b) to stochastic online scheduling. To describe the algorithm, we need the notion of α-points. For $0 < \alpha \leq 1$, the α-point $t_j(\alpha)$ of a job j is the first moment in time when an α-fraction of j has been completed in the LP schedule. α-points were introduced by Sousa (1989), and have since been used in the design of a variety of approximation and online algorithms for scheduling problems.

The algorithm that we analyze here works as follows. It maintains, side-by-side with the actual schedule on m machines, the preemptive LP schedule on the (virtual) single machine. For this, we create a priority list L of jobs, sorted by nonincreasing ratios of weight to expected processing time. Initially, L is empty. Whenever a new job j arrives, we draw some value $\alpha_j \in (0, 1]$ uniformly at random (independent from the drawings for other jobs). Moreover, job j is inserted into the correct position in L. (Ties are broken arbitrarily.) We obtain the LP schedule by scheduling, at any point in time, the first job in L on the virtual machine. As soon as job j has reached its α_j-point in the LP schedule; i.e., when it has been processed for $\alpha_j p_j$ units of time on the virtual machine, it is randomly assigned to one of the m machines (independent of the assignments of other jobs). It then enters another priority list on that machine, which is arranged by nondecreasing α-points. (As before, ties are broken arbitrarily.) On each real machine, jobs are then scheduled nonpreemptively in that order. Note that, by construction, no job can start before its α-point. Finally, whenever a job j is completed on the virtual machine (i.e., it has been processed for p_j periods of time), it is removed from L. And when it is completed on its real machine, it is deleted from the priority list of that machine. This concludes the description of the algorithm, which we call *RSOS* (for Randomized Stochastic Online Scheduling).

Theorem 3. *Let I be an instance of the stochastic scheduling problem to minimize the total weighted completion time on identical parallel machines in which jobs arrive over time, and let Δ be an upper bound on the squared coefficients of variation of the jobs' processing times. Moreover, let $OPT(I)$ be the objective function value of an optimal off-line nonanticipative scheduling policy for I. Finally, let $RSOS(I)$ be the value of the schedule produced by the randomized online policy RSOS. Then, $E[RSOS(I)] \leq (\Delta + 2)E[OPT(I)]$.*

Proof. Let us consider an arbitrary, but fixed job j. Initially, let us also fix the index i of the machine to which j has been assigned, as well as a value of α_j. Note that j is ready to start at time $t_j(\alpha_j)$ on machine i; in particular, $r_j \leq t_j(\alpha_j)$. If j is not started at time $t_j(\alpha_j)$, then it is delayed by jobs with a smaller α-point that have been assigned to the same machine i. We denote by $E_{i,\alpha_j}[C_j]$ the conditional expected completion time of job j, where the expectation is taken both over the random choices of the algorithm, except for i and α_j, which are still fixed, and the processing times. We then have

$$\begin{aligned} E_{i,\alpha_j}[C_j] &\leq t_j(\alpha_j) + \mu_j + \sum_{k \neq j} \mu_k \, P(k \text{ on } i \text{ before } j) \\ &\leq t_j(\alpha_j) + \mu_j + \sum_{k \neq j} \mu_k \frac{1}{m} \frac{1}{p_k} \int_0^{t_j(\alpha_j)} I_k(t)\, dt \\ &\leq t_j(\alpha_j) + \mu_j + t_j(\alpha_j) = 2\, t_j(\alpha_j) + \mu_j \ . \end{aligned}$$

In the first inequality, $P(k \text{ on } i \text{ before } j)$ is the probability that job $k \neq j$ is assigned to the same machine as j and will be started before j. The probability that k is assigned to machine i is $1/m$. The integral in the second inequality captures the fraction of job k that is processed in the LP schedule before $t_j(\alpha_j)$, which, by the choice of α_k, is precisely the probability of $t_k(\alpha_k)$ being smaller than $t_j(\alpha_j)$. The remaining two inequalities are straightforward. We finally get rid of the conditional expectation by noting that the average α_j-point is equal to the mean busy time M_j in the LP schedule (Goemans et al. 2002). Therefore,

$$E[C_j] \leq 2 \int_0^1 t_j(\alpha_j)\, d\alpha_j + \mu_j = 2\, M_j + \mu_j \ .$$

The result now follows from our earlier observation that $\sum_{j\in N} w_j M_j - \frac{\Delta-1}{2}\sum_{j\in N} w_j \mu_j$ is a lower bound on the expected value of an optimal policy (Lemma 1), and so is $\sum_{j\in N} w_j\mu_j$. Hence,

$$\begin{aligned} E\Big[\sum_{j\in N} w_j C_j\Big] &\leq 2\sum_{j\in N} w_j M_j + \sum_{j\in N} w_j\mu_j \\ &= 2\sum_{j\in N} w_j M_j - (\Delta-1)\sum_{j\in N} w_j\mu_j + \Delta\sum_{j\in N} w_j\mu_j \\ &\leq (2+\Delta)\mathrm{OPT}(I) \ . \end{aligned}$$ □

The crucial observation, which makes this proof work, is that the set of jobs that is scheduled on machine i before job j does not depend on the actual realization of processing times. The order of jobs is determined by the LP schedule, which depends only on the expected processing times.

Corollary 4. *There exists a deterministic $(2+\Delta)$-approximation algorithm for the stochastic (off-line) problem of minimizing the weighted sum of completion times on identical parallel machines subject to release dates.*

We omit the proof from this extended abstract, but note that this algorithm can be obtained from RSOS by the method of conditional probabilities (Spencer 1987). Of course, this implies that the derived algorithm does not work in an online context. This will be fixed, to some extent, in the next section.

4 A Deterministic Algorithm

A simple, though somewhat less effective way of derandomizing the RSOS policy, yet one that does not destroy its online nature, is to choose α_j deterministically beforehand. The rest of the algorithm, to which we will refer as *DSOS*, remains unchanged, except that jobs are not randomly assigned to machines. Instead, we employ a list scheduling strategy: whenever a machine becomes available, we start a job with the smallest α-point of all not-yet-processed jobs whose α-points have already passed. Let ϕ denote the golden ratio, and let us choose $\alpha_j = \phi - 1$ for all $j \in N$.

Theorem 5. *Let I be an instance of the stochastic scheduling problem to minimize the total weighted completion time on identical parallel machines in which jobs arrive over time, and let Δ be an upper bound on the squared coefficients of variation of the jobs' processing times. Moreover, let $OPT(I)$ be the objective function value of an optimal off-line nonanticipative scheduling policy for I. Finally, let $DSOS(I)$ be the value of the schedule produced by the deterministic online policy DSOS. Then, $E[DSOS(I)] \leq \max\{\phi + 1, \frac{\phi+1}{2}\Delta + \frac{\phi+3}{2}\}E[OPT(I)]$.*

Proof. The proof is saved for the full version of this paper; it is based on careful modifications of the proof of Correa and Wagner (2008, Theorem 3.2), which itself is based on that of Goemans et al. (2002, Theorem 3.3). Apart from Lemma 1, the key insight is that the start of any job j is always delayed by the same set of jobs, regardless of the actual instantiation of processing times. $\square$

5 Concluding Remarks

In this paper, we have taken the design and analysis of approximation and online algorithms for nonpreemptive stochastic scheduling problems a step further. The main ingredient is a new linear programming relaxation for stochastic scheduling problems on identical parallel machines that is provably stronger than the one that was used in the design of all previously proposed approximate policies.

While the algorithms studied here do have deterministic counterparts that were analyzed before, it is important to recognize that the extension of algorithms designed for deterministic scheduling problems to stochastic problems is not automatic. In fact, many approaches that work well for deterministic scheduling problems cannot be modified to handle random processing times.

In the course of this work, we have obtained the first randomized policy for stochastic online scheduling as well as the best performance guarantee for stochastic (off-line) scheduling on identical parallel machines with release dates. Looking beyond the realm of stochastic scheduling, this paper provides additional proof of the versatility of the LP schedule, which had previously been used to derive a series of best known performance guarantees, competitive ratios, and asymptotic optimality results for a variety of scheduling problems; see Goemans et al. (2002), Schulz and Skutella (20002a, 20002b), Chou et al. (2006), and Correa and Wagner (2008), among others.

It is also worth mentioning that both RSOS and DSOS need information on expected processing times only, even though they are compared to optimal policies that have full access to the entire distribution. Moreover, as was the case for the previous linear programming relaxation by Möhring, Schulz, and Uetz (1999), the new linear programming relaxation remains valid for preemptive schedules. In particular, the nonpreemptive schedules generated by the algorithms considered here and in previous papers are approximate solutions for preemptive stochastic scheduling as well. While their performance guarantees are not as good as the one in Megow and Vredeveld (2006), the policies are simple, can be implemented in polynomial time, and require little information about the distribution of processing times.

Acknowledgments

The author is grateful to the organizers of Dagstuhl Seminar 05031, where this work was originally conceived, and to José Correa for sending him a preliminary version of Correa and Wager (2008), which inspired him to revisit stochastic scheduling problems.

References

Afrati, F., Bampis, E., Chekuri, C., Karger, D., Kenyon, C., Khanna, S., Milis, I., Queyranne, M., Skutella, M., Stein, C., Sviridenko, M.: Approximation schemes for minimizing average weighted completion time with release dates. In: Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science, pp. 32–43 (1999)

Borodin, A., El-Yaniv, R.: Online Computation and Competitive Analysis. Cambridge University Press, Cambridge (1998)

Chakrabarti, S., Phillips, C., Schulz, A., Shmoys, D., Stein, C., Wein, J.: Improved scheduling algorithms for minsum criteria. In: auf der Heide, F., Monien, B. (eds.) ICALP 1996. LNCS, vol. 1099, pp. 646–657. Springer, Heidelberg (1996)

Chekuri, C., Motwani, R., Natarajan, B., Stein, C.: Approximation techniques for average completion time scheduling. SIAM Journal on Computing 31, 146–166 (2001)

Chou, M., Liu, H., Queyranne, M., Simchi-Levi, D.: On the asymptotic optimality of a simple on-line algorithm for the stochastic single-machine weighted completion time problem and its extensions. Operations Research 54, 464–474 (2006)

Chou, M., Queyranne, M., Simchi-Levi, D.: The asymptotic performance ratio of an on-line algorithm for uniform parallel machine scheduling with release dates. Mathematical Programming 106, 137–157 (2006)

Correa, J., Wagner, M.: LP-based online scheduling: From single to parallel machines. Mathematical Programming (in press, 2008)

Goemans, M., Queyranne, M., Schulz, A., Skutella, M., Wang, Y.: Single machine scheduling with release dates. SIAM Journal on Discrete Mathematics 15, 165–192 (2002)

Hall, L., Schulz, A., Shmoys, D., Wein, J.: Scheduling to minimize average completion time: Off-line and on-line approximation algorithms. Mathematics of Operations Research 22, 513–544 (1997)

Hall, W., Wellner, J.: Mean residual life. In: Csörgö, M., Dawson, D., Rao, J., Saleh, A.E. (eds.) Proceedings of the International Symposium on Statistics and Related Topics, pp. 169–184 (1981)

Kämpke, T.: On the optimality of static priority policies in stochastic scheduling on parallel machines. Journal of Applied Probability 24, 430–448 (1987)

Lenstra, J., Rinnooy Kan, A., Brucker, P.: Complexity of machine scheduling problems. Annals of Discrete Mathematics 1, 343–362 (1977)

Megow, N., Schulz, A.: On-line scheduling to minimize average completion time revisited. Operations Research Letters 32, 485–490 (2004)

Megow, N., Uetz, M., Vredeveld, T.: Models and algorithms for stochastic online scheduling. Mathematics of Operations Research 31, 513–525 (2006)

Megow, N., Vredeveld, T.: Approximation results for preemptive stochastic online scheduling. In: Azar, Y., Erlebach, T. (eds.) ESA 2006. LNCS, vol. 4168, pp. 516–527. Springer, Heidelberg (2006)

Möhring, R., Radermacher, F., Weiss, G.: Stochastic scheduling problems I: General strategies. Zeitschrift für Operations Research 28, 193–260 (1984)

Möhring, R., Schulz, A., Uetz, M.: Approximation in stochastic scheduling: The power of LP-based priority policies. Journal of the ACM 46, 924–942 (1999)

Queyranne, M., Schulz, A.: Scheduling unit jobs with compatible release dates on parallel machines with nonstationary speeds. In: Balas, E., Clausen, J. (eds.) IPCO 1995. LNCS, vol. 920, pp. 307–320. Springer, Heidelberg (1995)

Rothkopf, M.: Scheduling with random service times. Management Science 12, 703–713 (1966)

Schulz, A., Skutella, M.: The power of α-points in preemptive single machine scheduling. Journal of Scheduling 5, 121–133 (2002a)

Schulz, A., Skutella, M.: Scheduling unrelated machines by randomized rounding. SIAM Journal on Discrete Mathematics 15, 450–469 (2002b)

Sgall, J.: On-line scheduling. In: Fiat, A., Woeginger, G. (eds.) Online Algorithms: The State of the Art. LNCS, vol. 1442, ch. 9, pp. 196–231. Springer, Heidelberg (1998)

Skutella, M., Uetz, M.: Stochastic machine scheduling with precedence constraints. SIAM Journal on Computing 34, 788–802 (2005)

Sousa, J.: Time Indexed Formulations of Non-Preemptive Single-Machine Scheduling Problems. Ph.D. thesis, Université Catholique de Louvain, Belgium (1989)

Spencer, J.: Ten Lectures on the Probabilistic Method. CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 52. SIAM, Philadelphia (1987)

Weber, R., Varaiya, P., Walrand, J.: Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flowtime. Journal of Applied Probability 23, 841–847 (1986)

Weiss, G.: Approximation results in parallel machines stochastic scheduling. Annals of Operations Research 26, 195–242 (1990)

Weiss, G., Pinedo, M.: Scheduling tasks with exponential service times on nonidentical processors to minimize various cost functions. Journal of Applied Probability 17, 187–202 (1980)