Approximation in Stochastic Scheduling: The Power of LP-based Priority Policies

Rolf H. Möhring
Technische Universität Berlin, Germany
Andreas S. Schulz
M.I.T., Cambridge, Massachusetts, USA
and
Marc Uetz
Technische Universität Berlin, Germany

We consider the problem to minimize the total weighted completion time of a set of jobs with individual release dates which have to be scheduled on identical parallel machines. Job processing times are not known in advance, they are realized on-line according to given probability distributions. The aim is to £nd a scheduling policy that minimizes the objective in expectation. Motivated by the success of LP-based approaches to deterministic scheduling, we present a polyhedral relaxation of the performance space of stochastic parallel machine scheduling. This relaxation extends earlier relaxations that have been used, among others, by Hall, Schulz, Shmoys, and Wein [1997] in the deterministic setting. We then derive constant performance guarantees for priority policies which are guided by optimum LP solutions, and thereby generalize previous results from deterministic scheduling. In the absence of release dates, the LP-based analysis also yields an additive performance guarantee for the WSEPT rule which implies both a worst-case performance ratio and a result on its asymptotic optimality, thus complementing previous work by Weiss [1990]. The corresponding LP lower bound generalizes a previous lower bound from deterministic scheduling due to Eastman, Even, and Isaacs [1964], and exhibits a relation between parallel machine problems and corresponding problems with only one fast single machine. Finally, we show that all employed LPs can be solved in polynomial time by purely combinatorial algorithms.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Sequencing and scheduling*; G.2.1 [**Discrete Mathematics**]: Combinatorics—*Combinatorial algorithms*; G.1.6 [**Numerical Analysis**]: Optimization—*Linear Programming*; F.1.2 [**Computation by Abstract Devices**]: Modes of Computation—*Online computation*

General Terms: ALGORITHMS, THEORY

Additional Key Words and Phrases: Stochastic scheduling, Approximation, Worst-case performance, Priority policy, LP-relaxation, WSEPT rule, Asymptotic optimality

---

This research was partially supported by the German-Israeli Foundation for Scienti£c Research and Development (G.I.F.) under grant I 246-304.02/97. An extended abstract appeared in the Proceedings of the 2nd Int. Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'99).
Authors' addresses: Rolf H. Möhring and Marc Uetz. Technische Universität Berlin, Fachbereich Mathematik, Sekr. MA 6–1, Straße des 17. Juni 136, 10623 Berlin, Germany, Email: {moehring, uetz}@math.tu–berlin.de. Andreas S. Schulz. MIT, Sloan School of Management and Operations Research Center, E53–361, 30 Wadsworth St, Cambridge, MA 02139, Email: schulz@mit.edu.

1. INTRODUCTION

In the past years, LP-based approximation techniques have evolved impressively. These methods have been successfully applied to a variety of combinatorial optimization problems, including scheduling problems. Most efforts have concentrated on deterministic models, and quite often results on their stochastic counterparts involve very specialized techniques. With this work we intend to show that, to a certain extent, polyhedral methods also carry over to the algorithm design and analysis of stochastic scheduling problems.

The model. Let $J = \{1, \ldots, n\}$ be a set of jobs which have to be non-preemptively scheduled on m identical parallel machines so as to minimize the total weighted completion time. That is, each job has a nonnegative weight w_j and one wants to minimize $\sum_{j \in J} w_j C_j$, where C_j denotes the completion time of job j. Any machine can process at most one job at a time, and every job has to be processed on one of the m machines. We consider scenarios where jobs may, or may not have individual release dates $r_j \geq 0$. The crucial assumption is that processing times of jobs are not known in advance, but are instead given by a random variable $\boldsymbol{p} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n)$. Here, $\boldsymbol{p}_j$ denotes the random variable for the processing time of job j. (All random variables are typeset in bold face.) Throughout the paper job durations are supposed to be stochastically independent, and £rst as well as second moments are £nite. It is usually assumed that these distributions are known from the outset, but for our approach it suf£ces that the expected processing times (and an upper bound on their coef£cients of variation) are given. Using the well known classi£cation scheme for scheduling problems introduced by Graham, Lawler, Lenstra, and Rinnooy Kan [1979], the problem under consideration may be written as $\mathrm{P} \mid \boldsymbol{p}_j \sim \text{stoch}, r_j \mid E[\sum w_j \boldsymbol{C}_j]$.

Due to the lack of beforehand information on processing times, the jobs have to be allocated to machines "on-line". This dynamic allocation of jobs to machines is the task of a *scheduling policy*. It speci£es which job(s) should be started at any given time t. The decisions of such a policy may only depend on the "past up to time t", which is given by the sets of jobs already £nished or being performed at t, their start times, and the conditional distribution of remaining processing times of jobs. In other words, it is required that a policy does not anticipate future information. Within the framework of stochastic dynamic optimization this is known as the *non-anticipative* character of policies. For a detailed account of the theoretical foundations of the stochastic model considered in this paper, particularly the characterization of policies, we refer the reader to [Möhring et al. 1984, 1985]. Of special importance for our work is the class of *priority policies*, which implement a given priority order on the set of jobs; they will be formally de£ned in Section 4.

A given policy Π eventually results in a feasible schedule for any vector of "a posteriori" realized processing times. Hence, it associates with every vector p of possible processing times a vector $S^\Pi(p)$ of feasible start times:

$$\mathbb{R}_+^n \ni (p_1, \ldots, p_n) = p \overset{\Pi}{\longmapsto} S^\Pi(p) = (S_1^\Pi(p), \ldots, S_n^\Pi(p)) \in \mathbb{R}_+^n.$$

Simple examples show that in general one cannot expect to £nd a non-anticipative scheduling policy that minimizes the objective point-wise for any realization of processing times. Therefore, one aims to minimize the objective in expectation. If $E[\boldsymbol{C}_j^\Pi]$ denotes the expected completion time of job j when scheduling according to policy Π, one can formulate

the problem as

$$\text{minimize } \{\sum_{j\in J} w_j E[\boldsymbol{C}_j^{\Pi}] \mid \Pi \text{ policy}\}.$$

Furthermore, we let

$$Z^{OPT} := \inf\{\sum_{j\in J} w_j E[\boldsymbol{C}_j^{\Pi}] \mid \Pi \text{ policy }\}$$

denote the corresponding optimum value. It follows from [Möhring et al. 1984, Section 4] that in the present setting there exists an *optimum policy* with expected performance Z^{OPT}. Note that an optimum policy is not necessarily work conserving. It may involve deliberate idling of machines, even in the absence of job release dates.

The model considered in this paper is somewhat related to certain on-line scenarios, which recently have received quite some attention. These scenarios are also based on the assumption that the scheduler does not have access to the whole instance at once, but rather learns the input piece by piece over time and has to make decisions based on partial knowledge only. When carried to an extreme, there is both a lack of knowledge on jobs arriving in the future and the running time of every job is unknown until it completes. In contrast to the stochastic model introduced above, on-line algorithms are usually analyzed with respect to optimum off-line solutions. We refer to [Sgall 1998] for an overview of recent achievements in this direction. Note that stochastic scheduling is also more moderate than on-line scheduling in the sense that one supposes that the number of jobs to be scheduled as well as (at least) their expected job processing times are known in advance. Our approach also differs from the probabilistic analysis of parallel machine scheduling problems as considered, e.g., by Spaccamela, Rhee, Stougie, and van de Geer [1992] or Chan, Muriel, and Simchi-Levi [1998], where it is assumed that the whole instance, including processing times of jobs, is known in advance.

Related work. Stochastic machine scheduling problems have been considered, among others, by Glazebrook [1979], Weiss and Pinedo [1980], Bruno, Downey, and Frederickson [1981], Möhring, Radermacher, and Weiss [1984, 1985], Weber, Varaiya, and Walrand [1986], Kämpke [1987], and Weiss [1990, 1992]. For a survey and more bibliographic references we refer to Section 16 of the survey by Lawler, Lenstra, Rinnooy Kan, and Shmoys [1993]. Except for the mentioned work of Möhring, Radermacher, and Weiss [1984, 1985] and Weiss [1990, 1992], research mainly concentrated on identifying conditions that guarantee optimality of simple priority policies such as SEPT, LEPT (shortest/longest expected processing times £rst), or WSEPT (schedule jobs with highest ratio of weight to expected processing time £rst). Already for the deterministic case without release dates, the problem under consideration is NP-hard, even for £xed $m \geq 2$ [Bruno et al. 1974], and the WSPT rule (weighted shortest processing time £rst) is known to achieve a worst-case performance ratio of $\frac{1}{2}(\sqrt{2}+1)$ [Kawaguchi and Kyan 1986]. For the special case of a single machine, WSPT is known to be optimal [Smith 1956], and this result easily generalizes to stochastic processing times [Rothkopf 1966]. However, results for parallel machines are more complex. For unit weights, the SEPT rule is optimal whenever job processing times are exponentially distributed [Weiss and Pinedo 1980] or, more generally, whenever the processing time distributions of the jobs are stochastically comparable in pairs [Weber et al. 1986], but it fails to be optimal in general. For arbitrary weights, the WSEPT rule is optimal whenever processing times are exponentially distributed and additionally the

job weights are compliant with the ratios of weight to expected processing time [Kämpke 1987]. In the general case, Weiss [1990, 1992] has analyzed the optimality gap of WSEPT, and he proved that WSEPT is asymptotically optimal under mild assumptions on the input parameters of the problem. To the best of our knowledge, no results were previously known for problems where jobs are released over time. Our work also relates to recent developments in the optimal control of stochastic systems [Bertsimas and Niño-Mora 1996; Glazebrook and Niño-Mora 1997; Dacre et al. 1999], and we will discuss similarities and differences in Section 4.4.

Results. Our approach to stochastic machine scheduling is LP-based, and motivated by the success of polyhedral approaches to deterministic scheduling problems. The driving idea is to exploit a polynomially solvable LP-relaxation of the performance space of the problem in order to get both a lower bound on the performance of an optimum policy as well as some guidance to design a corresponding LP-based priority policy with provably good performance. Most relevant for our work in this respect is the paper by Hall, Schulz, Shmoys, and Wein [1997], where several approximation algorithms are derived on the basis of LP-relaxations in completion time variables. For related and previous work in deterministic scheduling, we refer to the bibliographic references therein. We extend this methodology to the stochastic setting, and obtain constant performance guarantees for both the models with and without job release dates. For the model with release dates, we derive an LP-based priority policy with a performance guarantee of $3-\frac{1}{m}+\max\{1,\frac{m-1}{m}\Delta\}$, where Δ is an upper bound on the squared coef£cients of variation of the occurring probability distributions. The underlying polyhedral relaxations of the performance space generalize previous relaxations that have been used in the deterministic setting. Based on [Queyranne 1993], we further show that all employed LP-relaxations can be solved in polynomial time by purely combinatorial algorithms.

Apart from priority policies which are guided by optimum LP-solutions, we also analyze the performance of the WSEPT rule for the model without non-trivial job release dates, and we derive a worst-case performance guarantee of $1+\frac{(\Delta+1)(m-1)}{2m}$. Examples show that the performance ratio of $\frac{1}{2}(\sqrt{2}+1)$ of the WSPT rule in deterministic scheduling does not generalize to the stochastic setting. Furthermore, the LP-based analysis yields in fact an additive bound for the performance of WSEPT which implies its asymptotic optimality, thus complementing previous results by Weiss [1990]. The LP lower bound also generalizes a previous lower bound on the cost of any deterministic schedule by Eastman, Even, and Isaacs [1964]. One thus obtains a lower bound on the expected cost of any scheduling policy in terms of the optimum cost for a corresponding problem with only one fast single machine.

Organization of the paper. Section 2 introduces the basic concept of LP-based priority policies in stochastic scheduling, while in Section 3 a new class of valid inequalities for the performance space in stochastic parallel machine scheduling is presented. In Section 4.1, this polyhedral relaxation is used to prove a constant performance guarantee for an LP-based priority policy within the model where jobs may have non-trivial release dates. The analysis of the performance of WSEPT for the model without release dates is presented in Section 4.2. We conclude with some remarks in Section 5. The appendix provides purely combinatorial algorithms to solve the LP-relaxations used in Section 4.

2. LP-BASED APPROXIMATION IN STOCHASTIC SCHEDULING

A policy is called an α-*approximation* if its expected performance is within a factor of α of the optimum expected value, and if it can be determined and executed in polynomial time with respect to the input size of the problem. To cope with the input size of a stochastic scheduling problem, which includes non-discrete data in general, we assume that the input is speci£ed by the number of jobs, the number of machines, and the encoding lengths of weights w_j, release dates r_j, expected processing times $E[\boldsymbol{p}_j]$, and, as the sole stochastic information, an upper bound on the coef£cients of variation of all processing time distributions $\boldsymbol{p}_j$, $j = 1,\ldots,n$. The coef£cient of variation of a given random variable X is the ratio $\sqrt{\mathrm{Var}[X]}/E[X]$. Thus, it is particularly suf£cient if all second moments $E[\boldsymbol{p}_j^2]$ are given. This notion of input size is motivated by the fact that from a practitioner's point of view the expected processing times of jobs together with the assumption of some typical distribution "around them" is realistic and usually suf£ces to describe a stochastic scheduling problem. Note, however, that the performance guarantees we derive actually hold with respect to optimal policies that make use of the *complete* knowledge of the distributions of processing times.

In most cases optimal policies and the corresponding optimum value Z^{OPT} are unknown. Hence, in order to prove performance guarantees for simple priority policies we use lower bounds on the optimum value Z^{OPT}. The problem we consider can be written as

$$\text{minimize } \{ \sum_{j\in J} w_j C_j \mid C \in \mathcal{C} \},$$

where $\mathcal{C} := \{ (E[\boldsymbol{C}_1^\Pi],\ldots,E[\boldsymbol{C}_n^\Pi]) \mid \Pi \text{ policy } \} \subseteq \mathbb{R}_+^n$ denotes the *performance space*. Since one cannot hope to completely characterize the performance space in general, we approximate $\mathcal{C}$ by a polyhedron P which is de£ned by valid inequalities for $\mathcal{C}$. Thus $\mathcal{C} \subseteq P$. We then solve the LP relaxation

$$\text{minimize } \{ \sum_{j\in J} w_j C_j \mid C \in P \},$$

and denote by $C^{LP} = (C_1^{LP},\ldots,C_n^{LP})$ some optimal solution to this relaxation. If the LP captures suf£cient structure of the original problem, the ordering of jobs according to non-decreasing values of C_j^{LP} is a promising candidate for a priority policy (see Section 4 for a formal de£nition). If Π denotes such a policy, clearly

$$\sum_{j\in J} w_j C_j^{LP} \quad \leq \quad Z^{OPT} \quad \leq \quad \sum_{j\in J} w_j E[\boldsymbol{C}_j^\Pi],$$

and the goal is to prove $\sum_{j\in J} w_j E[\boldsymbol{C}_j^\Pi] \leq \alpha \sum_{j\in J} w_j C_j^{LP}$, for some $\alpha \geq 1$. This leads to a performance guarantee of α for the priority policy Π and also to a (dual) guarantee for the quality of the LP lower bound:

$$\sum_{j\in J} w_j E[\boldsymbol{C}_j^\Pi] \leq \alpha\, Z^{OPT} \qquad \text{and} \qquad \sum_{j\in J} w_j C_j^{LP} \geq \frac{1}{\alpha}\, Z^{OPT}.$$

3. VALID INEQUALITIES FOR STOCHASTIC PARALLEL MACHINE SCHEDULING

In deterministic scheduling, Schulz [1996, Lemma 7] proved that for any feasible schedule on m machines the following inequalities are valid:

$$\sum_{j\in A} p_j C_j \quad \geq \quad \frac{1}{2m}\Big(\sum_{j\in A} p_j\Big)^2 + \frac{1}{2}\sum_{j\in A} p_j^2 \qquad \text{for all } A \subseteq J. \tag{1}$$

Here, p_j and C_j denote the deterministic processing and completion times of jobs, respectively. The following class of valid inequalities extends (1) to stochastic parallel machine scheduling. They are crucial for all our subsequent results.

$$\begin{aligned}\sum_{j\in A} E[\boldsymbol{p}_j]\,E[\boldsymbol{C}_j^{\Pi}] \;\geq\; & \frac{1}{2m}\Big(\sum_{j\in A} E[\boldsymbol{p}_j]\Big)^2 + \frac{1}{2}\sum_{j\in A}(E[\boldsymbol{p}_j])^2 \\ & - \frac{m-1}{2m}\sum_{j\in A}\mathrm{Var}[\boldsymbol{p}_j] \qquad \text{for all } A \subseteq J.\end{aligned} \tag{2}$$

THEOREM 3.1. *Let* Π *be any policy for stochastic parallel machine scheduling. Then inequalities* (2) *are valid for the corresponding vector of expected completion times* $E[\boldsymbol{C}^{\Pi}]$.

PROOF. Consider any policy Π and any £xed realization p of processing times. Let $S_j := S_j^{\Pi}(p)$ denote the start time of job j subject to policy Π and p. Since $(S_1,\ldots,S_n)$ de£nes a feasible (deterministic) schedule for the given job durations $(p_1,\ldots,p_n)$, we may rewrite (1) to obtain

$$\sum_{j\in A} p_j S_j \;\geq\; \frac{1}{2m}\Big(\sum_{i,j\in A,\; i\neq j} p_i p_j\Big) \;-\; \frac{m-1}{2m}\sum_{j\in A} p_j^2 \;, \tag{3}$$

for any $A \subseteq J$. Now recall the connection between the distributions for processing, start, and completion times. Due to the non-anticipative character of policies and since processing times are independent, the random variables for the processing time $\boldsymbol{p}_j$ and the start time $\boldsymbol{S}_j^{\Pi}$ of any job j are stochastically independent. This yields in particular $E[\boldsymbol{p}_j \boldsymbol{S}_j^{\Pi}] = E[\boldsymbol{p}_j]E[\boldsymbol{S}_j^{\Pi}]$ for all $j \in J$ and all policies Π. Furthermore, recalling that $\mathrm{Var}[\boldsymbol{p}_j] = E[\boldsymbol{p}_j^2] - E[\boldsymbol{p}_j]^2$ and taking expectations in (3) yields:

$$\begin{aligned}\sum_{j\in A} E[\boldsymbol{p}_j]\,E[\boldsymbol{S}_j^{\Pi}] \;&\geq\; \frac{1}{2m}\Big(\sum_{i,j\in A,\; i\neq j} E[\boldsymbol{p}_i \boldsymbol{p}_j]\Big) \;-\; \frac{m-1}{2m}\sum_{j\in A} E[\boldsymbol{p}_j^2] \\ &=\; \frac{1}{2m}\Big(\sum_{i,j\in A,\; i\neq j} E[\boldsymbol{p}_i]\,E[\boldsymbol{p}_j]\Big) \;-\; \frac{m-1}{2m}\sum_{j\in A} E[\boldsymbol{p}_j^2] \\ &=\; \frac{1}{2m}\Big(\sum_{j\in A} E[\boldsymbol{p}_j]\Big)^2 \;-\; \frac{1}{2}\sum_{j\in A} E[\boldsymbol{p}_j]^2 \\ &\quad - \frac{m-1}{2m}\sum_{j\in A}\mathrm{Var}[\boldsymbol{p}_j] \qquad \text{for all } A \subseteq J.\end{aligned}$$

Now, $E[\boldsymbol{C}_j^{\Pi}] = E[\boldsymbol{S}_j^{\Pi}] + E[\boldsymbol{p}_j]$ concludes the proof. □

Weiss [1999] has communicated to us that an alternate proof of the validity of inequalities (2) can be obtained on the basis of [Weiss 1990], where an exact formula for the

left-hand side of (2) is derived for non-idling (i.e., work conserving) policies.

With an additional assumption on the second moments of all processing time distributions, one can rewrite (2) more conveniently. Therefore, assume that the squared coef£cients of variation of all processing times $\boldsymbol{p}_j$ are bounded by some constant Δ, that is,

$$\mathrm{Var}[\boldsymbol{p}_j]/(E[\boldsymbol{p}_j])^2 \le \Delta \qquad \text{for all jobs } j \in J. \tag{4}$$

Then, the following inequalities are valid for the performance space C:

$$\begin{aligned}\sum_{j\in A} E[\boldsymbol{p}_j]E[\boldsymbol{C}_j^{\Pi}] \;\ge\; & \frac{1}{2m}\left(\Big(\sum_{j\in A} E[\boldsymbol{p}_j]\Big)^2 + \sum_{j\in A} E[\boldsymbol{p}_j]^2\right) \\ & - \frac{(m-1)(\Delta-1)}{2m}\left(\sum_{j\in A} E[\boldsymbol{p}_j]^2\right) \quad \text{for all } A \subseteq J.\end{aligned} \tag{5}$$

COROLLARY 3.1. *Let Π be any policy for stochastic parallel machine scheduling. If $\mathrm{Var}[\boldsymbol{p}_j]/(E[\boldsymbol{p}_j])^2 \le \Delta$ for all processing time distributions $\boldsymbol{p}_j$, then inequalities* (5) *are valid for the corresponding vector of expected completion times $E[\boldsymbol{C}^{\Pi}]$.*

Note that an upper bound on the coef£cients of variation of the $\boldsymbol{p}_j$ is a quite natural assumption for scheduling problems. For instance, if job processing times follow NBUE distributions (i.e., the expected remaining processing time of a job in process never exceeds its total expected processing time), it follows from [Hall and Wellner 1981] that $\mathrm{Var}[\boldsymbol{p}_j]/(E[\boldsymbol{p}_j])^2 \le 1$.

4. CONSTANT PERFORMANCE GUARANTEES FOR (LP-BASED) PRIORITY POLICIES IN STOCHASTIC MACHINE SCHEDULING

In this section, we derive constant worst-case performance guarantees for LP-based priority policies in stochastic machine scheduling.

Let us £rst give a formal de£nition of priority policies. A job j is called *available* at time t if $r_j \le t$, and if all its predecessors have already been completed by time t (in the case that precedence constraints are also given). A policy is called a *priority policy* or *priority rule* or *list scheduling policy* if at any time t a maximal number of available jobs is scheduled according to a given priority order on the set of jobs. More precisely, we are given a linear order on J, and when a machine is or becomes idle at time t, the available job with highest priority is started at t. Widely used priority policies are, e.g., LEPT and SEPT as well as WSEPT.

In the presence of release dates or precedence constraints, a priority policy may schedule jobs with low priority prior to jobs with higher priority. If this is not desired, we additionally enforce that jobs with low priority are scheduled only if all jobs with higher priority have already been started. In this case, we call a job j available at time t if $r_j \le t$ and all its predecessors with respect to the given priority order have already been started. The corresponding priority policy is then called *job-based*. Note that this may yield idling of machines although there are jobs waiting that in principle could have been started.

4.1 Parallel machine scheduling with release dates

We now consider the problem $\mathrm{P} \mid \boldsymbol{p}_j \sim \text{stoch}, \; r_j \mid E[\sum w_j \boldsymbol{C}_j]$. The £rst ingredient in our development of a near-optimal policy is an upper bound on the expected completion times whenever the jobs are scheduled according to a (job-based) priority policy. The following

lemma is a generalization of a corresponding bound for the deterministic case [Phillips et al. 1998; Hall et al. 1997]. For the deterministic case without release dates, a similar bound already appears in [Eastman et al. 1964].

LEMMA 4.1. *Let* Π *be a job-based priority policy which schedules the jobs in the order* $1 < \cdots < n$ *. Then,*

$$E[\boldsymbol{C}_j^{\Pi}] \;\le\; \max_{k=1,\ldots,j} r_k \;+\; \frac{1}{m}\left(\sum_{k=1}^{j-1} E[\boldsymbol{p}_k]\right) \;+\; E[\boldsymbol{p}_j] \quad \text{for all } j \in J. \tag{6}$$

PROOF. Consider any job j, and a corresponding policy Π^j that starts the £rst job at time $\max_{k=1,\ldots,j} r_k$ and proceeds in the same order as Π. Then jobs $1,\ldots,j-1$ are scheduled without any inserted idle time in the order $1 < 2 < \cdots < j-1$, and j starts as soon as a machine becomes available. Now let p be any realization of processing times. Policy Π^j does not involve idle times between time $\max_{k=1,\ldots,j} r_k$ and the start of job j. Thus job j starts not later than $\max_{k=1,\ldots,j} r_k + \frac{1}{m}\sum_{k=1}^{j-1} p_k$ under policy Π^j, since at $\max_{k=1,\ldots,j} r_k$ the £rst job is started, and $\frac{1}{m}\sum_{k=1}^{j-1} p_k$ is the average load per machine with respect to jobs $1,\ldots,j-1$. Thus we have $C_j \le \max_{k=1,\ldots,j} r_k + \frac{1}{m}(\sum_{k=1}^{j-1} p_k) + p_j$ where $C_j = C_j^{\Pi^j}(p)$ is the completion time of j subject to Π^j and p. Now, if the jobs are scheduled according to Π, job j is scheduled at least as early as under policy Π^j, and this holds for any realization of processing times. Thus (6) even holds point-wise for Π, and taking expectations completes the proof. □

Note that in the above proof we crucially need to consider job-based priority policies instead of ordinary priority policies if release dates are present. In the absence of release dates, clearly $\max_{k=1,\ldots,j} r_k = 0$, and the claim also holds for ordinary priority policies.

The second ingredient establishes the critical linkage between the LP solution and the value obtained from an LP-based priority policy; it is again a generalization of a corresponding result in deterministic scheduling [Hall et al. 1997; Schulz 1996].

LEMMA 4.2. *Let* $m \ge 1$ *and* $C \in \mathbb{R}^n$ *be any point which satis£es* $C_j \ge E[\boldsymbol{p}_j]$ *for all* $j \in J$ *as well as inequalities* (5) *for some* $\Delta \ge 0$*. Assume without loss of generality that* $C_1 \le \cdots \le C_n$*, then*

$$\frac{1}{m}\sum_{k=1}^{j} E[\boldsymbol{p}_k] \;\le\; \left(1 \;+\; \max\{1, \frac{m-1}{m}\Delta\}\right) C_j \;\text{ for all }\; j = 1,\ldots,n.$$

PROOF. Consider any set $\{1,\ldots,j\}$, $j \in J$. Then, due to inequalities (5) and due to the fact that $C_j \ge \cdots \ge C_1$,

$$C_j \sum_{k=1}^{j} E[\boldsymbol{p}_k] \ge \sum_{k=1}^{j} E[\boldsymbol{p}_k] C_k \ge \frac{1}{2m}\Big(\sum_{k=1}^{j} E[\boldsymbol{p}_k]\Big)^2 + \frac{m-\Delta(m-1)}{2m}\sum_{k=1}^{j} E[\boldsymbol{p}_k]^2.$$

We divide by $\sum_{k=1}^{j} E[\boldsymbol{p}_k]$ to obtain

$$C_j \ge \frac{1}{2m}\sum_{k=1}^{j} E[\boldsymbol{p}_k] \;+\; \frac{m-\Delta(m-1)}{2m} \cdot \frac{\sum_{k=1}^{j} E[\boldsymbol{p}_k]^2}{\sum_{k=1}^{j} E[\boldsymbol{p}_k]}.$$

Now consider the case $\Delta \le \frac{m}{m-1}$. Then the last term is nonnegative, and thus $\frac{1}{m}\sum_{k=1}^{j} E[\boldsymbol{p}_k] \le 2C_j$. For the case $\Delta \ge \frac{m}{m-1}$ the last term is nonpositive. But since

$$C_j \ge \max_{k=1,\dots,j} E[\boldsymbol{p}_k] \ge \frac{\sum_{k=1}^{j} E[\boldsymbol{p}_k]^2}{\sum_{k=1}^{j} E[\boldsymbol{p}_k]},$$

we obtain $C_j \ge \frac{1}{2m}\sum_{k=1}^{j} E[\boldsymbol{p}_k] + \frac{m-\Delta(m-1)}{2m} C_j$. The claim follows. □

We are now ready to analyze the following LP-based approximation algorithm for stochastic parallel machine scheduling with release dates. Suppose that the squared coef£cient of variation of processing times is bounded from above by some $\Delta \ge 0$. Then inequalities (5) are valid for any scheduling policy Π. Moreover, every vector of expected completion times corresponding to Π additionally ful£lls

$$E[\boldsymbol{C}_j^{\Pi}] \ge r_j + E[\boldsymbol{p}_j] \qquad \text{for all } j \in J. \tag{7}$$

We thus consider the linear programming relaxation

$$\min\{ \sum_{j\in J} w_j C_j \mid (5) \text{ and } (7) \}, \tag{8}$$

and let C^{LP} denote an optimum solution to (8). De£ne Π to be a job-based priority policy according to the order given by nondecreasing values of C_j^{LP}.

THEOREM 4.1. *Let* $Var[\boldsymbol{p}_j]/(E[\boldsymbol{p}_j])^2 \le \Delta$ *for all jobs* j *and some* $\Delta \ge 0$, *and let* Π *be the job-based priority policy corresponding to an optimal solution to the linear programming relaxation* (8). *Then* Π *is a* $(3 - \frac{1}{m} + \max\{1, \frac{m-1}{m}\Delta\})$*–approximation.*

In Appendix B we show that linear program (8) can be solved in $O(n^2)$ time by purely combinatorial methods. This implies that the corresponding priority order can be computed ef£ciently.

PROOF. First assume without loss of generality that $C_1^{LP} \le C_2^{LP} \le \cdots \le C_n^{LP}$. We apply Lemma 4.1 to Π, and observe that $\max_{k=1,\dots,j} r_k \le C_j^{LP}$ for all $j = 1,\dots,n$. This holds, since from inequalities (7) we get $C_k^{LP} \ge r_k$, and because $C_j^{LP} \ge C_{j-1}^{LP} \ge \cdots \ge C_1^{LP}$. Moreover, $E[\boldsymbol{p}_j] \le C_j^{LP}$, thus Lemma 4.1 yields

$$E[\boldsymbol{C}_j^{\Pi}] \le \left(2 - \frac{1}{m}\right) C_j^{LP} + \frac{1}{m}\left(\sum_{k=1}^{j} E[\boldsymbol{p}_k]\right)$$

for all jobs $j \in J$. Since C^{LP} ful£lls the conditions of Lemma 4.2, we now obtain

$$E[\boldsymbol{C}_j^{\Pi}] \le \left(3 - \frac{1}{m} + \max\{1, \frac{m-1}{m}\Delta\}\right) C_j^{LP}$$

for all jobs $j \in J$. The fact that linear program (8) is a relaxation of the scheduling problem concludes the proof. □

Theorem 4.1 particularly yields a worst-case performance guarantee of $(4 - \frac{1}{m})$ whenever $\mathrm{Var}[\boldsymbol{p}_j]/(E[\boldsymbol{p}_j])^2 \le m/(m-1)$ for the given processing time distributions. This bound is already known for deterministic scheduling [Hall et al. 1997].

4.2 Parallel machine scheduling without release dates

We now consider the problem P $\mid \boldsymbol{p}_j \sim$ stoch $\mid E[\sum w_j \boldsymbol{C}_j]$. Using the framework of the preceding section, one easily obtains an LP-based priority policy which has a performance guarantee of $2 - \frac{1}{m} + \max\{1, \frac{m-1}{m}\Delta\}$. However, for this case we can improve the result by considering the WSEPT rule and a different LP-relaxation which allows us to explicitly exploit the structure of an optimum LP solution within the analysis. Recall that WSEPT works as follows: When a machine becomes available, schedule the job(s) with highest ratio $w_j/E[\boldsymbol{p}_j]$ among the jobs not yet started.

THEOREM 4.2. *Let $Var[\boldsymbol{p}_j]/(E[\boldsymbol{p}_j])^2 \le \Delta$ for all jobs j and some $\Delta \ge 0$. Then the WSEPT priority policy is a $(1 + \frac{(\Delta+1)(m-1)}{2m})$–approximation.*

PROOF. First assume without loss of generality that $w_1/E[\boldsymbol{p}_1] \ge w_2/E[\boldsymbol{p}_2] \ge \cdots \ge w_n/E[\boldsymbol{p}_n]$. Now consider the linear programming relaxation

$$\min\{ \sum_{j\in J} w_j C_j \mid (5) \}, \tag{9}$$

and let C^{LP} denote an optimum solution with optimum value Z^{LP}. Since inequalities (5) de£ne a supermodular polyhedron, the solution to the LP-relaxation (9) is given by Edmonds' greedy algorithm for supermodular polyhedra (see Appendix A for details). Hence,

$$C_j^{LP} = \frac{1}{m}\sum_{k=1}^{j} E[\boldsymbol{p}_k] \; - \; \frac{(\Delta-1)(m-1)}{2m} E[\boldsymbol{p}_j] \quad \text{for } j = 1,\ldots,n.$$

We now apply Lemma 4.1 to the WSEPT priority policy to obtain

$$\begin{aligned} E[\boldsymbol{C}_j^{WSEPT}] \; &\le \; \frac{1}{m}\sum_{k=1}^{j} E[\boldsymbol{p}_k] \; + \; (1-\frac{1}{m}) E[\boldsymbol{p}_j] \\ &= \; C_j^{LP} \; + \; \frac{(\Delta+1)(m-1)}{2m} E[\boldsymbol{p}_j]. \end{aligned}$$

Since linear program (9) is a relaxation for the scheduling problem, and since $\sum_{j\in J} w_j E[\boldsymbol{p}_j]$ is a lower bound on the optimum value Z^{OPT}, we get

$$\begin{aligned} Z^{WSEPT} \; = \; \sum_{j\in J} w_j E[\boldsymbol{C}_j^{WSEPT}] \; &\le \; \sum_{j\in J} w_j C_j^{LP} \; + \; \frac{(\Delta+1)(m-1)}{2m} \sum_{j\in J} w_j E[p_j] \\ &\le \; Z^{LP} \; + \; \frac{(\Delta+1)(m-1)}{2m} Z^{OPT} \\ &\le \; \left(1 \; + \; \frac{(\Delta+1)(m-1)}{2m}\right) Z^{OPT}. \end{aligned}$$

□

It is clear from the proof of Theorem 4.2 that apart from the above worst-case ratio an additive performance guarantee for WSEPT can be derived as well.

COROLLARY 4.1. *Let $Var[\boldsymbol{p}_j]/(E[\boldsymbol{p}_j])^2 \le \Delta$ for all jobs j and some $\Delta \ge 0$, then*

$$Z^{WSEPT} \; - \; Z^{OPT} \; \le \; \frac{(\Delta+1)(m-1)}{2m} \sum_{j\in J} w_j E[p_j] \; .$$

Moreover, with some additional conditions on weights and expected processing times of the jobs, we obtain asymptotic optimality for the performance of the WSEPT rule.

COROLLARY 4.2. *If* $Var[\boldsymbol{p}_j]/(E[\boldsymbol{p}_j])^2 \leq \Delta$ *for all jobs* j *and some* $0 \leq \Delta < \infty$, *and if there exists some* $\varepsilon > 0$ *such that* $\varepsilon \leq w_j \leq 1/\varepsilon$ *and* $\varepsilon \leq E[p_j] \leq 1/\varepsilon$ *for all* j, *and if* $m/n \xrightarrow{n\to\infty} 0$, *then*

$$(Z^{WSEPT} - Z^{OPT})/Z^{OPT} \xrightarrow{n\to\infty} 0.$$

PROOF. First suppose without loss of generality that $w_1/E[\boldsymbol{p}_1] \geq w_2/E[\boldsymbol{p}_2] \geq \cdots \geq w_n/E[\boldsymbol{p}_n]$. Now let $Z_n^{OPT} := \sum_{j\in J} w_j E[p_j]$, and $Z_1^{OPT} := \sum_{j=1}^{n} w_j \sum_{k=1}^{j} E[p_k]$. Note that Z_1^{OPT} is the optimum value for a single machine problem, since the optimum policy on a single machine is WSEPT [Rothkopf 1966], and Z_n^{OPT} is the optimum value on n machines. Corollary 4.1 together with $Z^{OPT} \geq Z^{LP}$ now yields

$$(Z^{WSEPT} - Z^{OPT})/Z^{OPT} \leq \frac{(\Delta+1)(m-1)}{2m} \cdot \frac{Z_n^{OPT}}{Z^{LP}}.$$

But $Z^{LP} = \frac{1}{m} Z_1^{OPT} - \frac{(\Delta-1)(m-1)}{2m} Z_n^{OPT}$, thus the asymptotic behavior depends on the ratio $m Z_n^{OPT}/Z_1^{OPT}$. Under the condition that weights and expected processing times are bounded, this ratio is of order m/n. □

Similar considerations show that, subject to the same conditions, the LP-relaxation (9) is also asymptotically tight.

Corollary 4.1 complements a previous result by Weiss [1990, 1992], who showed that

$$Z^{WSEPT} - Z^{OPT} \leq \frac{m-1}{2} \cdot \max_{j=1,\ldots,n} \frac{w_j}{E[\boldsymbol{p}_j]} \cdot \Omega.$$

Here, Ω is an upper bound on the second moment of the remaining processing time of any uncompleted job at any given point in time. With assumptions on the input parameters of the problem which assure that the right-hand side remains bounded, Weiss [1990] has thus proved asymptotic optimality of WSEPT for a wide class of processing time distributions. In fact, since one can construct examples which show that neither of the two above additive bounds dominates the other, Corollary 4.1 complements Weiss' analysis of the quality of the WSEPT rule in stochastic machine scheduling.

Theorem 4.2 also implies a performance guarantee of $\frac{3}{2} - \frac{1}{2m}$ for the WSPT rule in deterministic scheduling. This result can alternatively be derived using the bounds by Eastman, Even, and Isaacs [1964]. They have proved that the cost of any schedule in deterministic scheduling is bounded from below as $Z_m^{OPT} \geq \frac{1}{m} Z_1^{OPT} + \frac{m-1}{2m} \sum_{j=1}^{n} w_j p_j$, where Z_m^{OPT} denotes the optimum value on m parallel machines and Z_1^{OPT} is the optimum value for the same jobs on a single machine (which is induced by WSPT [Smith 1956]). Moreover, they have derived a matching upper bound for WSPT, namely $Z_m^{OPT} \leq Z^{WSPT} \leq \frac{1}{m} Z_1^{OPT} + \frac{m-1}{m} \sum_{j=1}^{n} w_j p_j$, which yields that WSPT has a worst-case performance guarantee of $\frac{3}{2} - \frac{1}{2m}$. However, their lower bound as well as the corresponding performance guarantee does not hold in the stochastic setting, as will become clear in Example 4.1 below. In fact, Kawaguchi and Kyan [1986] showed that the worst-case performance ratio of WSPT in the deterministic setting is exactly $\frac{1}{2}(\sqrt{2}+1)$. Again, their techniques do not apply if processing times are stochastic, and Example 4.1 reveals that their worst-case bound does not hold in this case either.

EXAMPLE 4.1. *Consider a set of four jobs $J = \{1,\dots,4\}$ which have to be scheduled on $m = 2$ machines. All jobs have weight 1, i.e., the objective is the total expected completion time $\sum_{j=1}^{4} E[C_j]$. Let $0 < \varepsilon < 1$. Jobs 1 and 2 have processing time ε with probability $1-\varepsilon$ and processing time $1/\varepsilon$ with probability ε, independent of each other. Then the expected processing time of these jobs is $1+\varepsilon-\varepsilon^2$, which we choose to be the deterministic processing time of jobs 3 and 4.*

Since all jobs have the same expected processing time, the expected total completion time on a single machine is $Z_1^{OPT} = 10$ for $\varepsilon \to 0$ for any priority policy. For the parallel (two) machine case, elementary calculations show that the optimum policy is to schedule according to the priority list $1 < 2 < 3 < 4$ if ε is small enough, and we obtain an expected total completion time of $Z_m^{OPT} = 4$ for $\varepsilon \to 0$. Thus, in sharp contrast to the deterministic model and the above mentioned bound by Eastman, Even, and Isaacs [1964], we obtain $\frac{1}{m} Z_1^{OPT} > Z_m^{OPT}$ for this example.

Moreover, since all jobs have identical expected processing times, any priority policy is SEPT (or WSEPT) in this example. Scheduling according to the priority list $3 < 4 < 1 < 2$, yields an expected total completion time of 6 for $\varepsilon \to 0$. This shows that SEPT (or WSEPT) may differ from the optimum value by a factor arbitrarily close to $\frac{3}{2}$, and the deterministic worst case bounds $\frac{3}{2} - \frac{1}{2m}$ and, a fortiori, $\frac{1}{2}(\sqrt{2}+1)$ for WSPT do not hold in the stochastic setting. □

However, the proof of Theorem 4.2 yields the following generalization of the lower bound by Eastman, Even, and Isaacs [1964] to stochastic machine scheduling.

COROLLARY 4.3. *If $Var[p_j]/(E[p_j])^2 \le \Delta$ for all processing times p_j, then*

$$Z_m^{OPT} \;\ge\; \frac{1}{m} Z_1^{OPT} \;-\; \frac{(\Delta-1)(m-1)}{2m} \sum_{j=1}^{n} w_j E[p_j], \tag{10}$$

where Z_m^{OPT} denotes the optimum value for a parallel machine problem on m machines, and Z_1^{OPT} is the optimum value of the same instance on a single machine.

PROOF. Again, let without loss of generality $w_1/E[p_1] \ge w_2/E[p_2] \ge \cdots \ge w_n/E[p_n]$. Since $Z_1^{OPT} = \sum_{j=1}^{n} w_j \sum_{k=1}^{j} E[p_k]$, the right-hand side of (10) is precisely the value of an optimal solution to the LP-relaxation (9), and this a lower bound on Z_m^{OPT}. □

This particularly shows that for $\Delta \le 1$ the optimum value for a single machine problem with an m-fold faster machine is a relaxation for the corresponding problem on m parallel machines. Moreover, Example 4.1 not only reveals that the condition $\Delta \le 1$ is necessary for the validity of the fast single-machine relaxation, but it also shows that — in contrast to the deterministic case — a negative term in the right-hand side of inequalities (2) is necessary as well.

4.3 The single machine case

In the single machine case, the proof of optimality for WSEPT dates back to 1966. It was presented by Rothkopf [1966], and the corresponding result in deterministic scheduling is due to Smith [1956]. Moreover, Queyranne [1993] has shown that in the deterministic case inequalities (1), for $m = 1$, provide a complete description of the convex hull of the performance space. Bertsimas and Niño-Mora [1996] extended this result to stochastic processing times. We note that for $m = 1$ both the optimality of WSEPT and the complete

polyhedral description of the performance space by inequalities (2) also follow from the analysis of the previous section.

We conclude this section with a remark on the approximability of more general stochastic single machine problems which also involve arbitrary precedence relations. Since for $m = 1$ inequalities (2) exactly correspond to the analogue in deterministic scheduling, we may use the same LP relaxations and arguments as in [Hall et al. 1997] to obtain the same performance guarantees for stochastic single machine problems. More precisely, the natural generalization of the techniques presented in [Hall et al. 1997; Schulz 1996] yields a priority policy which is a 2-approximation for $1 \mid \boldsymbol{p}_j \sim \text{stoch}, \text{ prec} \mid E[\sum w_j \boldsymbol{C}_j]$ and a job-based priority policy which is a 3-approximation for $1 \mid \boldsymbol{p}_j \sim \text{stoch}, r_j, \text{ prec} \mid E[\sum w_j \boldsymbol{C}_j]$. These results hold for arbitrary, independent processing time distributions.

4.4 LP-based priority policies and the achievable region approach to stochastic systems

The LP-based approach presented in this paper is closely related to recent developments in the optimal control of stochastic systems via characterizing or approximating "achievable regions". For instance, Bertsimas and Niño-Mora [1996] show that previous results on the optimality of Gittins indexing rules can alternatively be derived by a polyhedral characterization of corresponding performance spaces as (extended) polymatroids. Subsequently, Glazebrook and Niño-Mora [1997] have proved approximate optimality of Klimov's index rule in multiclass queueing networks with parallel servers. Their work is based on *approximate conservation laws* for the performance of Klimov's index rule (which corresponds to the WSEPT rule for the model we consider here). Since from the bounds (10) and (6) one can obtain an approximate conservation law for the performance of WSEPT, Theorem 4.2 (respectively Corollary 4.1) of the present paper can also be derived within their framework.

There is, however, an interesting difference between the techniques employed in their work and those of the present paper. For the case with non-trivial release dates (Section 4.1), we explicitly make use of an optimum *primal* solution of LP-relaxation (8) in order to obtain a priority policy with provably good performance. (Note that in this case the performance of WSEPT can be arbitrarily bad.) While the achievable region approach as proposed in [Glazebrook and Niño-Mora 1997] and [Dacre et al. 1999, Section 3] is also based on the concept of LP-relaxations, the *dual* of the corresponding LP-relaxation is solved in order to derive Klimov's index rule and to analyze its performance for the case of parallel servers. Primal and dual solutions, however, can in fact lead to substantially different priority policies.

5. CONCLUDING REMARKS

With this work we extend the concept of LP-based approximation algorithms from deterministic scheduling to a more general stochastic setting. Several previous deterministic results, including LP-relaxations for parallel machine scheduling and corresponding LP-based performance guarantees occur as special cases. For the model without release dates, our work complements previous work on the performance of the WSEPT rule, and extends a previous lower bound on the value of optimum schedules to the stochastic setting.

More generally, LP relaxations of scheduling problems are shown to be a quite powerful tool for producing not only good lower bounds, but also high-quality priority policies. It is one of the outcomes of our studies that successful combinatorial methods from determinis-

tic machine scheduling also bear on algorithm design and analysis for stochastic machine scheduling problems. Moreover, another advantage of using LP relaxations is that one not only obtains "a priori" worst-case bounds, but also "a posteriori" guarantees (by comparing the actual objective value and the LP bound) depending on the particular instance. This aspect adds to the practical appeal of this approach.

Altogether, the presented results underline the potential of the polyhedral approach to scheduling problems – in both the deterministic and the stochastic setting, and we hope that this methodology may also lead to progress in other stochastic systems besides scheduling.

APPENDIX

The appendix first provides the necessary details on supermodular polyhedra and Edmonds' greedy algorithm. We then show that LP relaxation (8) can be solved in polynomial time. This already follows from the supermodularity of the right-hand side of inequalities (5) via the ellipsoid method [Grötschel et al. 1988]. However, we give a purely combinatorial algorithm with running time $O(n^2)$. Notice that this algorithm is of interest in the deterministic case as well, since it turns some approximation algorithms presented in [Hall et al. 1997] (which so far relied on the ellipsoid method) into combinatorial algorithms.

A. SUPERMODULAR POLYHEDRA AND THE GREEDY ALGORITHM

A set function $f : 2^J \longrightarrow \mathbb{R}$ is called supermodular, if

$$f(A \cap B) + f(A \cup B) \geq f(A) + f(B) \quad \text{for all } A, B \subseteq J.$$

For a supermodular set function f with $f(\emptyset) = 0$, the polyhedron

$$P(f) := \{x \in \mathbb{R}^n \mid x(A) \geq f(A) \text{ for all } A \subseteq J\}$$

is called a *supermodular polyhedron*. Here, as usual $x(A) := \sum_{j \in A} x_j$ for $x \in \mathbb{R}^n$. If we let $a \in \mathbb{R}^n$ be strictly positive and $f : 2^J \longrightarrow \mathbb{R}$ be supermodular with $f(\emptyset) = 0$, then

$$P_a(f) := \{x \in \mathbb{R}^n \mid \sum_{j \in A} a_j x_j \geq f(A) \text{ for all } A \subseteq J\} \tag{11}$$

is a *linear transformation* of a supermodular polyhedron which we also call a supermodular polyhedron, for convenience. If we let $w_j \geq 0$ for $j \in J$, it is well known that linear optimization problems

$$\min\{\sum_{j \in J} w_j x_j \mid x \in P_a(f)\} \tag{12}$$

are solved by Edmonds' greedy algorithm [Edmonds 1970]. An optimal solution for (12) is then given by

$$x_j^* = \big(f(\{1,\ldots,j\}) - f(\{1,\ldots,j-1\})\big)/a_j \qquad \text{for } j = 1,\ldots,n,$$

where we assumed that $w_1/a_1 \geq w_2/a_2 \geq \cdots \geq w_n/a_n$, and we also used that $f(\{1,\ldots,0\}) = f(\emptyset) = 0$. Consequently, linear program (9) can in fact be solved in time $O(n \log n)$. We refer to the monograph of Fujishige [1991] for more details on supermodular polyhedra and their extensions.

B. ANALYSIS OF LINEAR PROGRAM (8)

To see that linear program (8) also £ts into the framework of supermodular polyhedra we need some preliminaries. We write linear program (8) as:

$$\min\{\sum_{j\in J} w_j x_j \mid \sum_{j\in A} E[\boldsymbol{p}_j] x_j \geq f(A)\ \forall\, A\subseteq J \text{ and } x_j \geq \ell_j\ \forall\ j\in J\}, \tag{13}$$

where $f(A) = \frac{1}{2m}(\sum_{j\in A} E[\boldsymbol{p}_j])^2 + \frac{m-\Delta(m-1)}{2m}\sum_{j\in A} E[\boldsymbol{p}_j]^2$ is the right-hand side of (5), and $\ell_j \geq 0$ are some nonnegative lower bounds on x_j, $j\in J$. For instance, in linear program (8) we have $\ell_j = r_j + E[\boldsymbol{p}_j]$.

Observe £rst that f is supermodular. According to the notation from de£nition (11), let $P_{E[\boldsymbol{p}]}(f)$ denote the polyhedron de£ned by inequalities $\sum_{j\in A} E[\boldsymbol{p}_j] x_j \geq f(A)$, $A\subseteq J$. Then, following Fujishige [1991, Section II.3.1], the polyhedron given by (13) is called the *reduction* of $P_{E[\boldsymbol{p}]}(f)$ by the vector $(\ell_1,\ldots,\ell_n)$. De£ne the auxiliary set function

$$\hat{f}(A) \;:=\; \max_{B\subseteq A}\{f(B) + \sum_{j\in A-B} E[\boldsymbol{p}_j]\,\ell_j\} \qquad \text{for all } A\subseteq J\,. \tag{14}$$

LEMMA B.1. *The set function $\hat{f}: 2^J \longrightarrow \mathbb{R}$ is supermodular. Furthermore, the reduction of $P_{E[\boldsymbol{p}]}(f)$ by vector ℓ is exactly given by $P_{E[\boldsymbol{p}]}(\hat{f})$ and is therefore again a supermodular polyhedron.*

For a proof, we refer to [Fujishige 1991, Theorem 3.3].

Thus, we may apply Edmonds' greedy algorithm to solve linear program (13). That is, if we assume without loss of generality that $w_1/E[\boldsymbol{p}_1] \geq w_2/E[\boldsymbol{p}_2] \geq \cdots \geq w_n/E[\boldsymbol{p}_n]$, an optimal solution to (13) is given by

$$x_k^* \;=\; \big(\hat{f}(\{1,\ldots,k\}) - \hat{f}(\{1,\ldots,k-1\})\big)/E[\boldsymbol{p}_k] \qquad \text{for } k=1,\ldots,n$$

where $\hat{f}(\{1,\ldots,0\}) = \hat{f}(\emptyset) = 0$. Consequently, the only remaining task is the computation of the values $\hat{f}(\{1\})$, $\hat{f}(\{1,2\})$, ..., $\hat{f}(J)$. To this end, note that

$$\hat{f}(A) = \sum_{j\in A} E[\boldsymbol{p}_j]\,\ell_j \;+\; \max_{B\subseteq A}\{\underbrace{f(B) - \sum_{j\in B} E[\boldsymbol{p}_j]\,\ell_j}_{=:\,\hat{g}(B)}\}, \qquad A\subseteq J.$$

Hence, the evaluation of $\hat{f}(A)$ for some $A\subseteq J$ results in a maximization problem of the set function $\hat{g}$ over the ground set A. Since $\hat{g}$ is again supermodular, its maximum can be determined in polynomial time with the help of the ellipsoid method [Grötschel et al. 1988]. However, in the remainder of this section we show how to compute the maximum in time $O(n\log n)$ by exploiting the special structure of $\hat{g}$. The ideas below are adapted from [Queyranne 1993, Section 5].

LEMMA B.2. *Let A^* be a set maximizing $\hat{g}(B)$, $B\subseteq A$, and let without loss of generality A^* be $\subseteq$-minimal. Then:*

$$k\in A^* \quad\Longleftrightarrow\quad \frac{1}{m}\sum_{j\in A^*} E[\boldsymbol{p}_j] \;>\; \frac{(\Delta-1)(m-1)}{2m} E[\boldsymbol{p}_k] + \ell_k.$$

PROOF. Let $k\in A^*$, then $\hat{g}(A^*) > \hat{g}(A^*\setminus\{k\})$ due to the de£nition of A^*. Elementary

calculations yield:

$$E[\boldsymbol{p}_k]\Big(\frac{1}{m}\sum_{j\in A^*}E[\boldsymbol{p}_j] \;-\frac{(\Delta-1)(m-1)}{2m}E[\boldsymbol{p}_k]-\;\ell_k\Big) = \hat{g}(A^*)-\hat{g}(A^*\setminus\{k\}) \;>\; 0,$$

and since $E[\boldsymbol{p}_k]>0$, the £rst claim follows.

For the reverse direction, let $\frac{1}{m}\sum_{j\in A^*}E[\boldsymbol{p}_j] > \frac{(\Delta-1)(m-1)}{2m}E[\boldsymbol{p}_k]+\ell_k$ and suppose $k\notin A^*$. But since

$$\begin{aligned}&\hat{g}(A^*\cup\{k\})-\hat{g}(A^*)\\ &\qquad = E[\boldsymbol{p}_k]\Big(\frac{1}{m}\sum_{j\in A^*}E[\boldsymbol{p}_j] \;-\frac{(\Delta-1)(m-1)}{2m}E[\boldsymbol{p}_k]-\;\ell_k\;+\frac{1}{m}E[\boldsymbol{p}_k]\Big) \;>\; 0,\end{aligned}$$

we have $\hat{g}(A^*\cup\{k\})>\hat{g}(A^*)$, a contradiction to the de£nition of A^*. □

Therefore we obtain the following result:

COROLLARY B.1. *Let A^* be a set maximizing $\hat{g}(B)$, for $B\subseteq A$. If $i\in A^*$ for some $i\in A$, we have $j\in A^*$ for every $j\in A$ with $\frac{(\Delta-1)(m-1)}{2m}E[\boldsymbol{p}_j]+\ell_j \le \frac{(\Delta-1)(m-1)}{2m}E[\boldsymbol{p}_i]+\ell_i$.*

Thus, in order to maximize $\hat{g}$ over some ground set A, we just sort the jobs $j\in A$ in nondecreasing order of $\frac{(\Delta-1)(m-1)}{2m}E[\boldsymbol{p}_j]+\ell_j$. Assume this order is given by $1,2,\ldots,|A|$, then A^* must be one of the nested sets $\emptyset$, $\{1\}$, $\{1,2\}$, $\ldots$, A. Consequently, the maximization problem for $\hat{g}$ can be solved in $O(n\log n)$ time. In fact, for $A=J$ this algorithm is an $O(n\log n)$ time separation algorithm for the polyhedron $P_{E[\boldsymbol{p}]}(f)$ and a given point $\ell\in\mathbb{R}^n$, since the calculation of $\max_{B\subseteq J}\hat{g}(B)$ exactly corresponds to the problem of £nding the most violated inequality from $\sum_{j\in B}E[\boldsymbol{p}_j]\,\ell_j \ge f(B)$, $B\subseteq J$.

Now recall that in order to solve linear program (13) we have to calculate a sequence of values $\hat{f}(\{1\})$, $\hat{f}(\{1,2\})$, $\ldots$, $\hat{f}(J)$. By virtue of Corollary B.1, it is not hard to see that this can be done in $O(n^2)$ total time, and thus we get the following result.

THEOREM B.1. *Linear program* (8) *can be solved in $O(n^2)$ time.*

ACKNOWLEDGMENTS

We are grateful to an anonymous referee and Maurice Queyranne for helpful comments that led to an improved presentation of the paper, and to Kevin Glazebrook and Gideon Weiss for stimulating discussions on a previous version of this paper [Möhring et al. 1998]. In particular, Kevin Glazebrook pointed out that our analysis also yields an additive bound for the WSEPT rule.

REFERENCES

BERTSIMAS, D. AND NIÑO-MORA, J. 1996. Conservation laws, extended polymatroids and multi-armed bandit problems: A polyhedral approach to indexable systems. *Mathematics of Operations Research 21*, 257–306.

BRUNO, J. L., COFFMAN JR., E. G., AND SETHI, R. 1974. Scheduling independent tasks to reduce mean £nishing time. *Communications of the Association for Computing Machinery 17*, 382 – 387.

BRUNO, J. L., DOWNEY, P. J., AND FREDERICKSON, G. N. 1981. Sequencing tasks with exponential service times to minimize the expected ¤owtime or makespan. *Journal of the Association for Computing Machinery 28*, 100–113.

CHAN, L. M. A., MURIEL, A., AND SIMCHI-LEVI, D. 1998. Parallel machine scheduling, linear programming, and parameter list scheduling heuristics. *Operations Research 46*, 5, 729–741.

DACRE, M., GLAZEBROOK, K. D., AND NIÑO-MORA, J. 1999. The achievable region approach to the optimal control of stochastic systems. *Journal of the Royal Statistical Society, Series B*. To appear.

EASTMAN, W. L., EVEN, S., AND ISAACS, I. M. 1964. Bounds for the optimal scheduling of *n* jobs on *m* processors. *Management Science 11*, 268–279.

EDMONDS, J. 1970. Submodular functions, matroids and certain polyhedra. In *Proceedings of the International Conference on Combinatorics, Calgary* (1970), pp. 69–87.

FUJISHIGE, S. 1991. *Submodular functions and optimization*, Volume 47 of *Annals of Discrete Mathematics*. North-Holland, Amsterdam.

GLAZEBROOK, K. D. 1979. Scheduling tasks with exponential service times on parallel machines. *Journal of Applied Probability 16*, 685–689.

GLAZEBROOK, K. D. AND NIÑO-MORA, J. 1997. Scheduling multiclass queueing networks on parallel servers: Approximate and heavy-traf£c optimality of Klimov's rule. In R. BURKARD AND G. WOEGINGER Eds., *Algorithms – ESA'97*, Volume 1284 of *Lecture Notes in Computer Science* (1997), pp. 232–245. Springer. Proceedings of the 5th Annual European Symposium on Algorithms, Graz.

GRAHAM, R. L., LAWLER, E. L., LENSTRA, J. K., AND RINNOOY KAN, A. H. G. 1979. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics 5*, 287–326.

GRÖTSCHEL, M., LOVÁSZ, L., AND SCHRIJVER, A. 1988. *Geometric algorithms and combinatorial optimization*, Volume 2 of *Algorithms and Combinatorics*. Springer, Berlin.

HALL, L. A., SCHULZ, A. S., SHMOYS, D. B., AND WEIN, J. 1997. Scheduling to minimize average completion time: Off-line and on-line approximation algorithms. *Mathematics of Operations Research 22*, 513–544.

HALL, W. J. AND WELLNER, J. A. 1981. Mean residual life. In M. CSÖRGÖ, D. A. DAWSON, J. N. K. RAO, AND A. K. MD. E. SALEH Eds., *Statistics and Related Topics* (1981), pp. 169–184. North-Holland. Proceedings of the International Symposium on Statistics and Related Topics, Ottawa.

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

KAWAGUCHI, T. AND KYAN, S. 1986. Worst case bound on an LRF schedule for the mean weighted ¤ow-time problem. *SIAM Journal on Computing 15*, 1119–1129.

LAWLER, E. L., LENSTRA, J. K., RINNOOY KAN, A. H. G., AND SHMOYS, D. B. 1993. Sequencing and scheduling: Algorithms and complexity. In *Logistics of Production and Inventory*, Volume 4 of *Handbooks in Operations Research and Management Science* (1993), pp. 445–522. North-Holland, Amsterdam.

MÖHRING, R. H., RADERMACHER, F. J., AND WEISS, G. 1984. Stochastic scheduling problems I: General strategies. *ZOR - Zeitschrift für Operations Research 28*, 193–260.

MÖHRING, R. H., RADERMACHER, F. J., AND WEISS, G. 1985. Stochastic scheduling problems II: Set strategies. *ZOR - Zeitschrift für Operations Research 29*, 65–104.

MÖHRING, R. H., SCHULZ, A. S., AND UETZ, M. 1998. Approximation in stochastic scheduling: The power of LP-based priority policies. Technical Report 595/1998, Department of Mathematics, Berlin University of Technology.

PHILLIPS, C. A., STEIN, C., AND WEIN, J. 1998. Minimizing average completion time in the presence of release dates. *Mathematical Programming 82*, 199–223.

QUEYRANNE, M. 1993. Structure of a simple scheduling polyhedron. *Mathematical Programming 58*, 263–285.

ROTHKOPF, M. H. 1966. Scheduling with random service times. *Management Science 12*, 703–713.

SCHULZ, A. S. 1996. Scheduling to minimize total weighted completion time: Performance guarantees of LP-based heuristics and lower bounds. In W. H. CUNNINGHAM, S. T. MCCORMICK, AND M. QUEYRANNE Eds., *Integer Programming and Combinatorial Optimization*, Volume 1084 of *Lecture Notes in Computer Science* (1996), pp. 301–315. Springer. Proceedings of the 5th International IPCO Conference, Vancouver.

SGALL, J. 1998. On-line scheduling. In A. FIAT AND G. J. WOEGINGER Eds., *Online Algorithms: The State of the Art*, Volume 1442 of *Lecture Notes in Computer Science* (1998), pp. 196–231. Springer.

Proceedings of the Dagstuhl Workshop on On-Line Algorithms.

SMITH, W. E. 1956. Various optimizers for single-stage production. *Naval Research and Logistics Quarterly 3*, 59–66.

SPACCAMELA, A. M., RHEE, W. S., STOUGIE, L., AND VAN DE GEER, S. 1992. Probabilistic analysis of the minimum weighted ¤owtime scheduling problem. *Operations Research Letters 11*, 67–71.

WEBER, R. R., VARAIYA, P., AND WALRAND, J. 1986. Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected ¤owtime. *Journal of Applied Probability 23*, 841–847.

WEISS, G. 1990. Approximation results in parallel machines stochastic scheduling. *Annals of Operations Research 26*, 195–242.

WEISS, G. 1992. Turnpike optimality of Smith's rule in parallel machines stochastic scheduling. *Mathematics of Operations Research 17*, 255–270.

WEISS, G. 1999. Personal communication.

WEISS, G. AND PINEDO, M. 1980. Scheduling tasks with exponential service times on non-identical processors to minimize various cost functions. *Journal of Applied Probability 17*, 187–202.