Theory Comput. Systems **39**, 121–136 (2006)
DOI: 10.1007/s00224-005-1261-z

Theory of Computing Systems

The Expected Competitive Ratio for Weighted Completion Time Scheduling

Alexander Souza and Angelika Steger

Institute of Theoretical Computer Science, ETH Zürich,
Universitätstr. 6, CH-8092 Zürich, Switzerland
{steger,asouza}@inf.ethz.ch

Abstract. A set of n independent jobs is to be scheduled without preemption on m identical parallel machines. For each job j, a diffuse adversary chooses the distribution F_j of the random processing time P_j from a certain class of distributions $\mathcal{F}_j$. The scheduler is given the expectation $\mu_j = \mathbb{E}[P_j]$, but the actual duration is not known in advance. A positive weight w_j is associated with each job j and all jobs are ready for execution at time zero. The scheduler determines a list of the jobs, which is then scheduled in a non-preemptive manner. The objective is to minimise the total weighted completion time $\sum_j w_j C_j$. The performance of an algorithm is measured with respect to the expected competitive ratio $\max_{F \in \mathcal{F}} \mathbb{E}[\sum_j w_j C_j / \text{OPT}]$, where C_j denotes the completion time of job j and OPT the offline optimum value.

We show a general bound on the expected competitive ratio for list scheduling algorithms, which holds for a class of so-called new-better-than-used processing time distributions. This class includes, among others, the exponential distribution.

As a special case, we consider the popular rule *weighted shortest expected processing time first* (WSEPT) in which jobs are processed according to the non-decreasing μ_j/w_j ratio. We show that it achieves $\mathbb{E}[\text{WSEPT}/\text{OPT}] \leq 3 - 1/m$ for exponential distributed processing times.

1. Introduction

Scheduling problems are very well studied combinatorial optimisation problems. Among others, the following *completion time scheduling* problem and its variants have attracted much attention. A set of jobs is to be processed on a set of machines. The objective function is to minimise the *total weighed completion time* $\sum_j w_j C_j$, where C_j denotes the time when job j is finished and w_j denotes a weight associated with job j.

In this paper we consider the *expected competitive ratio* of scheduling algorithms for stochastic variants of the problem. This measure is defined as the expectation (taken over all instances) of the objective-value achieved by an algorithm on a certain instance related to the optimum value of the same instance. One property of this measure is that it favours algorithms that perform well on "many" instances.

Previous Work. The deterministic version of the completion time scheduling problem has been studied intensively since the 1950s. For the weighted single-machine problem Smith [27] proved optimality of the so-called *weighted shortest processing time first* (WSPT) rule: schedule the jobs in order of the non-decreasing processing time and weight ratio. For the unweighted problem, i.e., all weights are equal to one, with m identical parallel machines, the optimality of the *shortest processing time first* (SPT) strategy was shown by Conway et al. [5].

In contrast, Bruno and Sethi [2] showed that the weighted problem with m parallel machines is already $\mathcal{NP}$-hard in the ordinary sense for constant m. However, Sahni [22] proved that it admits a fully polynomial time approximation scheme (FPTAS). If the number of machines is considered as part of the input, Lageweg and Lenstra [14] established that the weighted problem is $\mathcal{NP}$-hard in the strong sense (see also problem SS13 of [7]). An exact algorithm was given by Sahni [22]. Skutella and Woeginger [26] found a polynomial time approximation scheme (PTAS). Kawaguchi and Kyan [12] established that WSPT achieves $\frac{1}{2}(1+\sqrt{2})$ approximation ratio. Besides that, several constant factor approximations are known for variants of the problem, see, e.g., [16], [17], [24], [10], and [25].

Evidently, this problem was studied extensively from the worst-case perspective. However, a drawback of this approach is that it may be overly pessimistic: a scheduling algorithm with bad worst-case behaviour may perform rather well in practical applications. A natural step to overcome this problem is to consider stochastic scheduling, i.e., to interpret input data as random variables and to measure the performance of an algorithm ALG by its expected objective-value $\mathbb{E}[\text{ALG}]$.

In stochastic completion time scheduling, the scheduler is given the weight and expected processing time for each job and the objective function is to minimise the *expected* total weighted completion time $\mathbb{E}[\sum_j w_j C_j]$. Hence, an algorithm MIN is considered optimal with respect to that measure if it minimises the expected total weighted completion time over all algorithms.

Models that are $\mathcal{NP}$-hard in a deterministic setting sometimes allow a simple priority rule to be optimal for the probabilistic counterpart. For example, the rule *shortest expected processing time first* (SEPT), i.e., schedule jobs in order of non-decreasing expected processing times, is known to be optimal for many variants, see, e.g., [21], [29], [3], [11], and [28]. Moreover, for the weighted single-machine problem, the rule *weighted shortest expected processing time first* (WSEPT) is optimal [18]. WSEPT schedules the jobs in non-decreasing order of the expected processing time over weight ratio.

In their work Möhring et al. [15] proved very general bounds on LP-based algorithms for many stochastic completion time scheduling problems. Their method is based on LP-relaxations of a deterministic problem, where an optimum solution to this LP yields a lower bound for $\mathbb{E}[\text{MIN}]$. In addition, this solution also yields a priority rule which can be shown to be only a constant factor larger than $\mathbb{E}[\text{MIN}]$ (with mild assumptions on processing time distributions). Also additional constraints, e.g., release dates can be

handled with minor modifications to the LP. One of the main results is that for m parallel machines,

$$\frac{\mathbb{E}[\text{WSEPT}]}{\mathbb{E}[\text{MIN}]} \leq 2 - \frac{1}{m}$$

holds if processing times are drawn from distributions that are new-better-than-used in expectation (NBUE).

One property of the performance measure $\mathbb{E}[\text{ALG}]$ is that instances x with small value $\text{ALG}(x)$ tend to be neglected since they contribute little to the overall expected value. Hence, in this measure, algorithms are preferred that perform well on instances x with large optimum value $\text{OPT}(x)$. It depends on the application if such behaviour is desireable, but if one is interested in algorithms that perform well on "many" instances, this measure may seem inappropriate.

Let $\text{ALG}(x)$ denote the objective-value achieved by a certain algorithm and let $\text{OPT}(x)$ be the optimum value on instance x, then $\mathbb{E}[\text{ALG}/\text{OPT}]$ defines the *expected competitive ratio*, where the expectation is taken over all instances.

Regarding the above drawback, the measure $\mathbb{E}[\text{ALG}/\text{OPT}]$ seems to be interesting for the following intuition. The ratio $\text{ALG}(x)/\text{OPT}(x)$ relates the value of the objective function achieved by some algorithm ALG to the optimum OPT on the instance x. Thus, the algorithm is considered to perform well on instances that yield a small ratio, and bad on instances with a large ratio. Hence, if for "most" instances a small ratio is attained, the "few" instances with a large ratio will not increase the expectation drastically. See also [23] for a discussion.

Despite the vast literature on stochastic scheduling problems, it seems that the expected competitive ratio has only been considered in a paper by Coffman and Gilbert [4] and in the recent work of Scharbrodt et al. [23].

In [23] the *unweighted* completion time problem on parallel machines is considered for the SEPT rule. The main result is that the SEPT algorithm yields

$$\mathbb{E}\left[\frac{\text{SEPT}}{\text{OPT}}\right] = \mathcal{O}(1)$$

for identical parallel machines under relatively weak assumptions on job processing time distributions. Here SEPT denotes the objective-value achieved by the SEPT algorithm and OPT the objective-value of an *optimum offline* algorithm, i.e., an algorithm that is given the actual realisations of processing times. The general approach is to partition the probability space according to a series of "bad events". These events yield a series of bounds that give an estimate for the probability of "larger" values of SEPT/OPT.

In this paper we consider the *weighted* version of the completion time scheduling problem, prove a general bound on the expected competitive ratio, and analyse the WSEPT algorithm.

Model, Problem Definition, and Notation. Consider a set $J = \{1, 2, \ldots, n\}$ of n independent jobs that have to be scheduled non-preemptively on a set $M = \{1, 2, \ldots, m\}$ of m identical parallel machines. For each job j, a so-called *diffuse adversary* (see [13]) chooses the *distribution* F_j of the random *processing time* $P_j \geq 0$ out of a certain class of distributions $\mathcal{F}_j$. We assume that the processing times P_j are stochastically independent. The scheduler is given the expectation $\mu_j = \mathbb{E}[P_j]$ of each job j, but the actual

realisation p_j is only learned upon job completion. A positive *weight* w_j is associated with each job $j \in J$ and all jobs are ready for execution at time zero. Every machine can process at most one job at a time. Each job can be executed by any of the machines, but preemption and delays are not permitted. The *completion time* C_j of a job $j \in J$ is the latest point in time, such that a machine is busy processing the job.

In *list scheduling*, jobs are processed one-by-one according to a priority list. A so-called *online list scheduling algorithm* is given weight w_j and mean μ_j for all $j \in J$ and, based on that information, *deterministically* constructs a permutation π of J. This *list* π is then *scheduled* according to the following *policy*: whenever a machine is idle and the list is not empty, the job at the head of the list is removed and processed non-preemptively and without delay on the idle machine (with least index). Notice that the actual *realisations* of processing times are learned only upon job completion, i.e., the list is constructed *offline*, while the schedule is constructed *online*.

Once a realisation $p = (p_1, p_2, \ldots, p_n)$ of job processing times is fixed, this policy yields a realisation of the random variable $\text{TWC}(\pi) = \sum_{j \in J} w_j C_j$, which denotes the *total weighted completion time* for list π. Thus, for any realisation of job processing times, an *offline optimum list* π^* is defined by

$$\text{OPT}(p) = \text{TWC}(\pi^*) = \min\{\text{TWC}(\pi)\colon\ \pi \text{ is a permutation of } J\}. \tag{1}$$

This yields the random variable OPT of the minimum value of the objective function for the random processing time vector $P = (P_1, P_2, \ldots, P_n)$. Let ALG be an online list scheduling algorithm and let π denote the list produced by ALG on input $\mu = (\mu_1, \mu_2, \ldots, \mu_n)$ and $w = (w_1, w_2, \ldots, w_n)$. We define the random variable $\text{ALG} = \text{TWC}(\pi)$ as the total weighted completion time achieved by the algorithm ALG. It is important to note that any online list scheduling algorithm deterministically constructs one *fixed* list for all realisations, while the optimum list may be different for each realisation.

For any algorithm ALG, the ratio ALG/OPT defines a random variable that measures the relative performance of that algorithm compared with the offline optimum. We may thus define the *expected competitive ratio* of an algorithm ALG by

$$R(\text{ALG}, \mathcal{F}) = \max\left\{\mathbb{E}\left[\frac{\text{ALG}}{\text{OPT}}\right]\colon\ F \in \mathcal{F}\right\},$$

where the job processing time distributions $F = (F_1, F_2, \ldots, F_n)$ are chosen by a diffuse adversary from a class of distributions $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_n)$. The objective is to minimise the expected competitive ratio, and thus an algorithm is called *competitive optimal* if it yields this minimum over all algorithms.

In the standard classification scheme by Graham et al. [9] our completion time scheduling problem is denoted

$$P \mid P_j \sim F_j(\mu_j) \in \mathcal{F}_j \mid \sum_j w_j C_j.$$

The performance of an algorithm is measured in terms of expected competitive ratio $R(\text{ALG}, \mathcal{F})$.

This model can be seen as a hybrid between *stochastic scheduling* models and *competitive analysis*, since it comprises important aspects of them both.

In competitive analysis (see [1] for an introduction) the input for an algorithm becomes available only gradually and usually driven by an adversary. The *competitive ratio* is defined by the objective-value of a certain algorithm on a worst-case input compared with the offline-optimum, i.e., an optimum algorithm that sees the whole input in advance. An unrestricted adversary is usually considered overly powerful. One approach to limit its power is the *diffuse adversary* model introduced by Koutsoupias and Papadimitriou [13], where the adversary is allowed to choose the distribution of the input out of a certain class of distributions. As is done in competitive analysis, our model relates the performance of an algorithm to the offline optimum on each instance. However, rather than taking the maximum value of that ratio, we take the average over all instances weighted with a distribution specified by a diffuse adversary.

The similarities to classical stochastic scheduling are that processing times are drawn from a probability distribution, and that the number n of jobs, their weights w and most importantly their expected durations μ are known. The most important difference is that in stochastic scheduling the optimum is usually not defined as the offline optimum, but as an algorithm that, given w and μ only, minimises the expected total weighted completion time.

Results. We introduce the class of distributions that are new-better-than-used in expectation *relative* to a function h (NBUE_h). The $\mathrm{NBUE}_{\mathrm{OPT}}$ class comprises the exponential, geometric, and uniform distribution.

We allow the adversary to choose $\mathrm{NBUE}_{\mathrm{OPT}}$ processing time distributions and derive bounds to online list scheduling algorithms for the problem

$$P \mid P_j \sim F(\mu_j) \in \mathrm{NBUE}_{\mathrm{OPT}} \mid \sum_j w_j C_j,$$

where the performance of an algorithm is measured in terms of expected competitive ratio $R(\mathrm{ALG}, \mathrm{NBUE}_{\mathrm{OPT}})$.

Our analysis depends on a quantity α which is an upper bound of the probability that any pair of jobs is in the wrong order in a list of a certain online list algorithm ALG, compared with an optimum list. We would also like to point out that our analysis is significantly simpler compared with [23].

Theorem 3.2 states that $R(\mathrm{ALG}, \mathrm{NBUE}_{\mathrm{OPT}}) \leq 1/(1-\alpha)$ holds for the single-machine case. In Corollary 3.7 we show that $R(\mathrm{ALG}, \mathrm{NBUE}_{\mathrm{OPT}}) \leq 1/(1-\alpha)+1-1/m$ holds for m identical parallel machines.

These results reflect well the intuition that an algorithm should perform better, the smaller its probability of sequencing jobs in the wrong order.

As an important special case, Corollary 3.8 yields $\mathbb{E}[\mathrm{WSEPT}/\mathrm{OPT}] \leq 3 - 1/m$ for the WSEPT algorithm with m identical parallel machines and exponential distributed processing times. Simulations empirically demonstrate tightness of this bound.

2. New-Better-Than-Used Distributions

In this section we define a class of processing time distributions from which the diffuse adversary is allowed to choose. However, we first discuss the class of distributions

that are *new-better-than-used in expectation* (NBUE). The concept of NBUE random variables is well known in reliability theory [8], where it is considered as a relatively weak assumption. NBUE distributions are typically used to model the aging of system components, but have also proved useful in the context of stochastic scheduling. For the problem $P \mid \sqrt{\mathrm{Var}[P_j]} \leq \mathbb{E}[P_j] \mid \mathbb{E}[\sum_j w_j C_j]$ the bound $\mathbb{E}[\text{WSEPT}] \leq (2 - 1/m)\mathbb{E}[\text{MIN}]$ of Möhring et al. [15] holds for NBUE processing time distributions as an important special case. In addition, Pinedo and Weber [19] give bounds for shop scheduling problems assuming NBUE processing time distributions.

A random variable $X \geq 0$ is NBUE if $\mathbb{E}[X - t \mid X \geq t] \leq \mathbb{E}[X]$ holds for all $t \geq 0$, see, e.g., [8]. Examples of NBUE distributions are uniform, exponential, Erlang, geometric, and Weibull distribution (with shape parameter at least one).

Let X denote a random variable taking values in a set $V \subset \mathbb{R}_0^+$ and let $h(x) > 0$ be a real-valued function defined on V. The random variable $X \geq 0$ is *new-better-than-used in expectation relative to* h (NBUE_h) if

$$\mathbb{E}\left[\frac{X-t}{h(X)} \;\middle|\; X \geq t\right] \leq \mathbb{E}\left[\frac{X}{h(X)}\right] \tag{2}$$

holds for all $t \in V$, provided these expectations exist.

It is natural to extend the concept of NBUE_h distributions to functions h that have more than one variable. Let X denote a random variable taking values in a set $V \subset \mathbb{R}_0^+$, let $y \in W \subset \mathbb{R}^k$ for $k \in \mathbb{N}$, and let $h(x, y) > 0$ be a real-valued function defined on (V, W). The random variable $X \geq 0$ is NBUE_h if

$$\mathbb{E}\left[\frac{X-t}{h(X,y)} \;\middle|\; X \geq t\right] \leq \mathbb{E}\left[\frac{X}{h(X,y)}\right] \tag{3}$$

holds for all $t \in V$ and all $y \in W$, provided these expectations exist.

In what follows, the distribution function of a random variable X, is denoted by $F_X(t) = \Pr[X \leq t]$ and let $f_X(t) = (d/dt)F_X(t)$ denote its density. For any event A, let $F_{X|A}(t) = \Pr[X \leq t \mid A]$ and $f_{X|A}(t) = (d/dt)F_{X|A}(t)$ be the conditional distribution, and the conditional density of X given A, respectively. Now we establish several general properties of NBUE_h distributions.

Lemma 2.1. *Let X be* NBUE_h *and let Y be a random vector taking values in W independently of X, then*

$$\mathbb{E}\left[\frac{X-t}{h(X,Y)} \;\middle|\; X \geq t\right] \leq \mathbb{E}\left[\frac{X}{h(X,Y)}\right].$$

Proof. Since X is NBUE_h (3) yields

$$\begin{aligned}\mathbb{E}\left[\frac{X-t}{h(X,Y)} \;\middle|\; X \geq t\right] &= \int_{y\in W} \mathbb{E}\left[\frac{X-t}{h(X,y)} \;\middle|\; X \geq t\right] f_Y(y)\,dy \\ &\leq \int_{y\in W} \mathbb{E}\left[\frac{X}{h(X,y)}\right] f_Y(y)dy = \mathbb{E}\left[\frac{X}{h(X,Y)}\right],\end{aligned}$$

which completes the proof. □

Lemma 2.2. *Let X be* NBUE$_h$, $\alpha > 0$, *and let Y be a random vector taking values in W independently of X, then*

$$\mathbb{E}\left[\frac{\alpha X - t}{h(X, Y)} \,\middle|\, \alpha X \geq t\right] \leq \mathbb{E}\left[\frac{\alpha X}{h(X, Y)}\right]$$

holds for all $t \in V$.

Proof. Let $y \in W$ be fixed. As X is NBUE$_h$ one obtains that

$$\begin{aligned}\mathbb{E}\left[\frac{\alpha X - t}{h(X, y)} \,\middle|\, \alpha X \geq t\right] &= \alpha \mathbb{E}\left[\frac{X - t/\alpha}{h(X, y)} \,\middle|\, X \geq \frac{t}{\alpha}\right] \\ &\leq \alpha \mathbb{E}\left[\frac{X}{h(X, y)}\right] = \mathbb{E}\left[\frac{\alpha X}{h(X, y)}\right]\end{aligned}$$

for all $t \geq 0$. Taking the expectation of Y as in Lemma 2.1 completes the proof. □

Lemma 2.3. *Let X be* NBUE$_h$, *let Y be a random vector taking values in W independently of X, and let $g(y)$ be a function defined on W taking values in V, then*

$$\mathbb{E}\left[\frac{X - g(Y)}{h(X, Y)} \,\middle|\, X \geq g(Y)\right] \leq \mathbb{E}\left[\frac{X}{h(X, Y)}\right].$$

Proof. Since (3) holds for *all* $t \in V$ it holds especially for $t = g(y)$. Repeating the proof of Lemma 2.1 with this choice yields the claim. □

The next lemmas show that exponential, geometric, and uniform distributed random variables are NBUE$_h$ if h is a *non-decreasing* function in x, i.e., $h(x + t, y) \geq h(x, y)$ for all x, y and $t \geq 0$. We use $X \sim \text{Uni}(a, b)$, $X \sim \text{Exp}(\lambda)$, and $X \sim \text{Geo}(p)$ to denote that the distribution of the random variable X is uniform in $[a, b]$, exponential with parameter λ, and geometric with probability p, respectively.

Lemma 2.4. *If $X \sim \text{Exp}(\lambda)$ and $h(x, y) > 0$ is non-decreasing in x, then X is* NBUE$_h$.

Proof. For all $s \geq 0$ we have $f_{X|X\geq t}(t + s) = f_X(s)$ since X has memoryless density f_X. As h is non-decreasing in x it holds that $h(t+s, y) \geq h(s, y)$ for $t \geq 0$. We therefore obtain

$$\begin{aligned}\mathbb{E}\left[\frac{X - t}{h(X, y)} \,\middle|\, X \geq t\right] &= \int_{x=t}^{\infty} \frac{x - t}{h(x, y)} f_{X|X\geq t}(x)\, dx \\ &= \int_{s=0}^{\infty} \frac{t + s - t}{h(t + s, y)} f_{X|X\geq t}(t + s)\, ds \\ &\leq \int_{s=0}^{\infty} \frac{s}{h(s, y)} f_X(s)\, ds = \mathbb{E}\left[\frac{X}{h(X, y)}\right],\end{aligned}$$

which proves the lemma. □

Lemma 2.5. *If $X \sim \mathrm{Geo}(p)$ and $h(x, y) > 0$ is non-decreasing in x, then X is NBUE_h.*

Proof. The proof for the exponential distribution analogously carries over to the geometric distribution. □

Lemma 2.6. *If $X \sim \mathrm{Uni}(a, b)$ where $0 \leq a < b$ and $h(x, y) > 0$ is non-decreasing in x, then X is NBUE_h.*

Proof. We want to show that

$$\mathbb{E}\left[\frac{X-t}{h(X,y)} \;\middle|\; X \geq t\right] \leq \mathbb{E}\left[\frac{X}{h(X,y)}\right]$$

for all $t \in V = [a, b)$ and $y \in W$. Let the random variable $T \sim \mathrm{Uni}(a, b)$ be independent of X. Let $t \in [a, b)$ be arbitrary but fixed, and introduce the (dependent) random variable $S = ((b-t)/(b-a))(T-a)$ with distribution $S \sim \mathrm{Uni}(0, b-t)$. Observe that for $t \leq x \leq b$ we have $\Pr[X \leq x \mid X \geq t] = (x-t)/(b-t) = \Pr[S \leq x-t]$ and thus $f_{X|X\geq t}(x) = 1/(b-t) = f_S(x-t)$. Therefore

$$\begin{aligned}\mathbb{E}\left[\frac{X-t}{h(X,y)} \;\middle|\; X \geq t\right] &= \int_{x=t}^{b} \frac{x-t}{h(x,y)} \frac{1}{b-t}\, dx \\ &= \int_{s=0}^{b-t} \frac{s}{h(s+t,y)} \frac{1}{b-t}\, ds = \mathbb{E}\left[\frac{S}{h(S+t,y)}\right].\end{aligned}$$

With $a \geq 0$ and $(b-t)/(b-a) \leq 1$ we have $S \leq T$. A simple calculation shows that since $T < b$ and $a \leq t$ we have $T \leq ((b-t)/(b-a))(T-a) + t = S + t$. Hence, since h is non-decreasing we find $h(S+t, y) \geq h(T, y) > 0$ and we have

$$\mathbb{E}\left[\frac{S}{h(S+t,y)}\right] \leq \mathbb{E}\left[\frac{T}{h(T,y)}\right] = \mathbb{E}\left[\frac{X}{h(X,y)}\right],$$

which completes the proof since X and T are identically distributed. □

3. Weighted Completion Time Scheduling

Recall that the random variable OPT measures the value of the offline optimum of our problem. If we interpret OPT as a real-valued function defined on the set of processing time vectors, then $\mathrm{NBUE}_{\mathrm{OPT}}$ induces a class of distributions. In particular, we allow the diffuse adversary to choose $\mathrm{NBUE}_{\mathrm{OPT}}$ processing time distributions in the following way: all jobs fall into the same class, e.g., they are all exponential distributed, but the parameter, and thus the mean μ_j of each individual job j, is arbitrary. We denote this degree of freedom by $P_j \sim F(\mu_j) \in \mathrm{NBUE}_{\mathrm{OPT}}$. Hence we consider the problem

$$P \mid P_j \sim F(\mu_j) \in \mathrm{NBUE}_{\mathrm{OPT}} \mid \sum_j w_j C_j$$

for online list scheduling against a diffuse adversary. The performance of an algorithm is measured with respect to the expected competitive ratio $R(\text{ALG}, \text{NBUE}_{\text{OPT}})$. In Section 3.1 the single-machine case is studied, and the results are generalised to identical parallel machines in Section 3.2.

For all $j, k \in J$, we define the indicator variable $M_{j,k}$ for the event that the jobs j and k are scheduled on the same machine. It is easily observed that for any list π and job j the random completion time satisfies $C_j = \sum_{k \leq^\pi j} P_k M_{j,k}$, where $k \leq^\pi j$ denotes that job k is *not after* job j in the list π.

3.1. *Single-Machine Scheduling*

A list π is called a *weighted shortest processing time first* (WSPT) list (also known as Smith's ratio rule [18]) if the jobs are in non-decreasing order of processing time and weight ratio, i.e.,

$$\frac{p_j}{w_j} \leq \frac{p_k}{w_k} \qquad \text{for} \quad j \leq^\pi k. \tag{4}$$

It is a well-known fact in scheduling theory, see, e.g., [27], [18], that WSPT characterises the offline optimum for single-machine scheduling.

Bounding the Expected Competitve Ratio. Recall that $M_{j,k}$ takes the value one if jobs j and k are scheduled on the same machine, which is trivially true in single-machine scheduling. Thus, $\text{TWC}(\pi)$ can be rearranged to the more convenient form

$$\text{TWC}(\pi) = \sum_{j \in J} w_j C_j = \sum_{j \in J} w_j \sum_{k \leq^\pi j} P_k M_{j,k} = \sum_{j \in J} P_j \sum_{k \geq^\pi j} w_k.$$

We define the random variable $\Delta_{j,k} = w_k P_j - w_j P_k$ for all $j, k \in J$ and the indicator variable

$$X_{j,k} = \begin{cases} 1, & \text{if} \quad \Delta_{j,k} \geq 0 \quad \text{and} \quad k >^\pi j, \\ 0, & \text{otherwise,} \end{cases}$$

for any fixed list π. The intuition behind $X_{j,k}$ is that the variable takes the value one if the jobs j and k are scheduled in the wrong order in a list produced by an algorithm, compared with an optimum list. $\Delta_{j,k}$ measures the change of TWC if two consecutive jobs j and k within a list are swapped.

The random variable OPT is defined as the value of the offline optimum and $\text{TWC}(\pi)$ as the total weighted completion time of list π. The following relationship between the random variables OPT, $\text{TWC}(\pi)$, $\Delta_{j,k}$, and $X_{j,k}$ is one of the key connexions of our analysis.

Theorem 3.1. *For any list π it holds that*

$$\text{TWC}(\pi) = \text{OPT} + \sum_{j \in J} \sum_{k \geq^\pi j} \Delta_{j,k} X_{j,k}.$$

Proof. We prove that the claim holds for every realisation $p = (p_1, p_2, \ldots, p_n)$ of processing times and weights $w = (w_1, w_2, \ldots, w_n)$. Define

$$y_{j,k}(\pi) = \begin{cases} 1, & k >^{\pi} j, \\ 0, & \text{otherwise}, \end{cases}$$

where π is an arbitrary list. We use $y_{j,k} = y_{j,k}(\pi)$ and $y^*_{j,k} = y_{j,k}(\pi^*)$ as shorthand, where π^* is an optimum list for the outcome p.

Observe that $x_{j,k} = 1$ if and only if $y_{j,k} = 1$, and $y^*_{j,k} = 0$. To see this recall that $y^*_{j,k} = 0$, i.e., $k \leq^{\pi^*} j$ implies $p_k/w_k \leq p_j/w_j$ and hence $\delta_{j,k} \geq 0$. Further note that for $j \neq k$ we have $y_{j,k} = 1 - y_{k,j}$ and $y^*_{j,k} = 1 - y^*_{k,j}$.

For every list π it holds that

$$\text{TWC}(\pi) = \sum_{j \in J} \sum_{k \geq^{\pi} j} w_k p_j = \sum_{j \in J} \sum_{k \in J} w_k p_j y_{j,k}(\pi) + \sum_{j \in J} w_j p_j,$$

as already observed by Potts [20]. Now we calculate

$$\begin{aligned}
\text{TWC}(\pi) - \text{TWC}(\pi^*) &= \sum_{j \in J} \sum_{k \in J} w_k p_j (y_{j,k} - y^*_{j,k}) \\
&= \sum_{j \in J} \sum_{k >^{\pi} j} (w_k p_j (y_{j,k} - y^*_{j,k}) + w_j p_k (y_{k,j} - y^*_{k,j})) \\
&= \sum_{j \in J} \sum_{k >^{\pi} j} (w_k p_j (y_{j,k} - y^*_{j,k}) - w_j p_k ((1 - y_{k,j}) - (1 - y^*_{k,j}))) \\
&= \sum_{j \in J} \sum_{k >^{\pi} j} (w_k p_j (y_{j,k} - y^*_{j,k}) - w_j p_k (y_{j,k} - y^*_{j,k})) \\
&= \sum_{j \in J} \sum_{k >^{\pi} j} (w_k p_j - w_j p_k)(y_{j,k} - y^*_{j,k}) \\
&= \sum_{j \in J} \sum_{k >^{\pi} j} \delta_{j,k} (1 - y^*_{j,k}) = \sum_{j \in J} \sum_{k \geq^{\pi} j} \delta_{j,k} x_{j,k}
\end{aligned}$$

and the proof is complete. □

We are now in a position to prove our main result which is stated as follows.

Theorem 3.2. *Let ALG be any online list scheduling algorithm for*

$$1 \mid P_j \sim F(\mu_j) \in \text{NBUE}_{\text{OPT}} \mid \sum_j w_j C_j.$$

If $\Pr[X_{j,k} = 1] \leq \alpha < 1$ *holds for all* $j \leq^{\pi} k$ *in all ALG lists* π, *then*

$$R(\text{ALG}, \text{NBUE}_{\text{OPT}}) \leq \frac{1}{1 - \alpha}.$$

Proof. Let π denote the fixed ALG list for expected processing times μ and weights w. By Lemmas 2.3 and 2.2 we have that for $k >^{\pi} j$,

$$\mathbb{E}\left[\frac{\Delta_{j,k}}{\mathrm{OPT}} \;\middle|\; X_{j,k}=1\right] = \mathbb{E}\left[\frac{w_k P_j - w_j P_k}{\mathrm{OPT}} \;\middle|\; w_k P_j \geq w_j P_k\right] \leq \mathbb{E}\left[\frac{w_k P_j}{\mathrm{OPT}}\right] \tag{5}$$

holds for all $\mathrm{NBUE}_{\mathrm{OPT}}$ processing time distributions. Theorem 3.1 and linearity of expectation establish

$$\mathbb{E}\left[\frac{\mathrm{ALG}}{\mathrm{OPT}}\right] = \mathbb{E}\left[\frac{\mathrm{OPT} + \sum_{j\in J}\sum_{k\geq^{\pi} j}\Delta_{j,k}X_{j,k}}{\mathrm{OPT}}\right] = 1 + \sum_{j\in J}\sum_{k\geq^{\pi} j}\mathbb{E}\left[\frac{\Delta_{j,k}X_{j,k}}{\mathrm{OPT}}\right].$$

By conditioning on $X_{j,k}=1$, application of (5), and by $\Pr[X_{j,k}=1]\leq\alpha$ for all $j \leq^{\pi} k$ we obtain

$$\mathbb{E}\left[\frac{\mathrm{ALG}}{\mathrm{OPT}}\right] = 1 + \sum_{j\in J}\sum_{k\geq^{\pi} j}\Pr[X_{j,k}=1]\mathbb{E}\left[\frac{\Delta_{j,k}}{\mathrm{OPT}} \;\middle|\; X_{j,k}=1\right] \leq 1 + \alpha\left(\sum_{j\in J}\sum_{k\geq^{\pi} j}\mathbb{E}\left[\frac{w_k P_j}{\mathrm{OPT}}\right]\right) = 1 + \alpha\mathbb{E}\left[\frac{\mathrm{ALG}}{\mathrm{OPT}}\right].$$

Finally, rearranging the inequality and $\alpha < 1$ completes the proof. □

Analysis of the WSEPT Algorithm. Now we consider the popular WSEPT list scheduling algorithm, and calculate the expected competitve ratio for exponential distributed job processing times, i.e., the adversary commits to exponential distribution. In applications, processing times are often modelled by exponential distributed random variables. Hence this special case is rather important.

A list π is a WSEPT list, if scheduling is done according to a non-decreasing expected processing time and weight ratio, i.e.,

$$\frac{\mu_j}{w_j} \leq \frac{\mu_k}{w_k} \qquad \text{for} \quad j \leq^{\pi} k. \tag{6}$$

The random variable $\mathrm{WSEPT} = \mathrm{TWC}(\pi)$ defines the total weighted completion time for WSEPT lists π. Notice that WSEPT is an online list scheduling algorithm since its lists are determined with the knowledge of the weights and expected processing times, rather than their realisations.

Corollary 3.3. *The WSEPT algorithm for the stochastic scheduling problem* $1 \mid P_j \sim \text{Exp}(\mu_j^{-1}) \mid \sum_j w_j C_j$ *yields*

$$\mathbb{E}\left[\frac{\text{WSEPT}}{\text{OPT}}\right] \leq 2.$$

Proof. Observe that the function $\text{OPT}(p)$ is non-decreasing in p. Hence, by Lemma 2.4, exponential distributed random variables are NBUE_{OPT}. It is thus sufficient to prove $\Pr[X_{j,k} = 1] \leq \frac{1}{2}$ for $j \leq^{\pi} k$ in all WSEPT lists π. As $w_k P_j \sim \text{Exp}((w_k\mu_j)^{-1})$ and $w_j P_k \sim \text{Exp}((w_j\mu_k)^{-1})$ we have

$$\begin{aligned}\Pr[X_{j,k} = 1] &= \Pr[\Delta_{j,k} > 0] = \Pr[w_k P_j > w_j P_k] \\ &= \int_{t=0}^{\infty} \frac{e^{-t/w_j\mu_k}}{w_j\mu_k} \int_{s=t}^{\infty} \frac{e^{-s/w_k\mu_j}}{w_k\mu_j}\, ds\, dt = \frac{w_k\mu_j}{w_k\mu_j + w_j\mu_k} \leq \frac{1}{2}\end{aligned}$$

because $j \leq^{\pi} k$ implies $w_k\mu_j \leq w_j\mu_k$ by the WSEPT ordering (6). Application of Theorem 3.2 completes the proof. □

In order to examplify the theoretical result obtained in Corollary 3.3, experiments were run by simulating exponential distributed processing times. The proof of Corollary 3.3 indicates that maximum values of $\mathbb{E}[\text{WSEPT}/\text{OPT}]$ are to be expected if $w_k\mu_j = w_j\mu_k$ holds for all $j \leq^{\pi} k$. Hence we have chosen $\mu_j = w_j = 1$ and simulated the problem

$$1 \mid P_j \sim \text{Exp}(1) \mid \sum_j C_j.$$

Experiments were run on the instances

$$n = 2, 3, 4, 5, 10, 20, 30, 40, 50, 100, 200, 300, 400, 500, 1000,$$

where for each n the simulation was repeated 1000 times.

Table 1 depicts the results, where $R_{\min}$ denotes the minimum ratio WSEPT/OPT measured, $R_{\max}$ the maximum, and R_{avg} the average over the number of repetitions, respectively.

3.2. *Scheduling Identical Parallel Machines*

Now we generalise our results to online list scheduling on m identical parallel machines.

Let $\text{ALG}^{(\ell)}$ and $\text{OPT}^{(\ell)}$ denote the objective values achieved by the algorithm ALG and the offline optimum, respectively, on ℓ identical parallel machines. Moreover, the completion time vector for a list π on ℓ identical parallel machines is denoted by $C^{(\ell)}$.

Lemmas 3.4 and 3.5 are due to Eastman et al. [6] and reduce parallel-machine scheduling to single-machine scheduling. We have included the short proofs for the sake of completeness. The method has also proved useful in previous work, see, e.g., [10], [15], and [16].

Table 1. Experimental results.

n	$R_{\min}$	R_{avg}	$R_{\max}$
2	1.0000	1.2131	1.9974
3	1.0000	1.3692	2.8146
4	1.0000	1.4648	3.3045
5	1.0000	1.5390	3.9426
10	1.0336	1.7269	4.2513
20	1.1991	1.8531	3.4000
30	1.2338	1.8978	2.9656
40	1.4090	1.9235	3.3859
50	1.4407	1.9344	3.0831
100	1.5965	1.9606	2.6208
200	1.6982	1.9759	2.4156
300	1.7274	1.9866	2.2881
400	1.7896	1.9908	2.3472
500	1.8147	1.9919	2.2538
1000	1.8721	1.9962	2.1644

Lemma 3.4. *Let π be any job list for non-preemptive list scheduling and let P be processing times, then*

$$C_j^{(m)} \leq \frac{1}{m} C_j^{(1)} + \left(1 - \frac{1}{m}\right) P_j.$$

Proof. Without loss of generality, $\pi = (1, 2, \ldots, n)$, let $j \in J$ and define $J_j = \{1, 2, \ldots, j\}$. The jobs in J_{j-1} are started before job j, and j starts as soon as a machine becomes available.

Consider the schedule induced by J_{j-1} in which all jobs prior to j are scheduled. Since the schedule does not involve idle time and j is scheduled on machine i with the least total processing time so far, j is started prior to the average total processing time per machine, i.e.,

$$s_j^{(m)} = \sum_{k \leq^{\pi} j-1} p_k m_{i,k} \leq \frac{1}{m} \sum_{k \leq^{\pi} j-1} p_k = \frac{1}{m} s_j^{(1)}.$$

Now

$$c_j^{(m)} = s_j^{(m)} + p_j \leq \frac{1}{m} s_j^{(1)} + p_j = \frac{1}{m} c_j^{(1)} + \left(1 - \frac{1}{m}\right) p_j$$

completes the proof. □

Lemma 3.5. *For the scheduling problem $P \mid\mid \sum_j w_j C_j$ it holds that*

$$\text{OPT}^{(m)} \geq \frac{1}{m} \text{OPT}^{(1)}.$$

Proof. Let π^* denote the optimum list for the processing times p on m machines (which need not be the optimum list for one machine). The corresponding completion time vectors are denoted by $c^{(m)} = c(\pi^*)^{(m)}$ and $c^{(1)} = c(\pi^*)^{(1)}$, respectively. Without loss of generality, the jobs are indexed such that $c_1^{(m)} \leq c_2^{(m)} \leq \ldots \leq c_n^{(m)}$ holds. Consider the schedule induced by the subset $J_j = \{1, 2, \ldots, j\}$. Since j is the job in J_j to be completed last, the machine i on which it is scheduled is the one with the maximum total processing time in the schedule. Because machine i has at least average total processing time, it holds that

$$c_j^{(m)} = \sum_{k \leq^{\pi^*} j} p_k m_{i,k} \geq \frac{1}{m} \sum_{k \leq^{\pi^*} j} p_k = \frac{1}{m} c_j^{(1)}.$$

As π^* is the optimum list for p on m machines, and since this list may be used for single-machine scheduling, we have

$$\mathrm{OPT}(p)^{(m)} = \sum_{j \in J} w_j c_j^{(m)} \geq \frac{1}{m} \sum_{j \in J} w_j c_j^{(1)} \geq \frac{1}{m} \mathrm{OPT}(p)^{(1)},$$

which completes the proof. □

Theorem 3.6. *Let ALG be any online list scheduling algorithm for*

$$P \mid P_j \sim \mathrm{Stoch}(\mu_j) \mid \sum_j w_j C_j,$$

then

$$\mathbb{E}\left[\frac{\mathrm{ALG}^{(m)}}{\mathrm{OPT}^{(m)}}\right] \leq \mathbb{E}\left[\frac{\mathrm{ALG}^{(1)}}{\mathrm{OPT}^{(1)}}\right] + 1 - \frac{1}{m}.$$

Proof. Lemmas 3.4 and 3.5 establish

$$\frac{C_j^{(m)}}{\mathrm{OPT}^{(m)}} \leq \frac{C_j^{(1)}}{m\mathrm{OPT}^{(m)}} + \left(1 - \frac{1}{m}\right) \frac{P_j}{\mathrm{OPT}^{(m)}} \leq \frac{C_j^{(1)}}{\mathrm{OPT}^{(1)}} + \left(1 - \frac{1}{m}\right) \frac{P_j}{\mathrm{OPT}^{(m)}}.$$

Thus we have

$$\begin{aligned} \frac{\sum_{j \in J} w_j C_j^{(m)}}{\mathrm{OPT}^{(m)}} &\leq \frac{\sum_{j \in J} w_j C_j^{(1)}}{\mathrm{OPT}^{(1)}} + \left(1 - \frac{1}{m}\right) \frac{\sum_{j \in J} w_j P_j}{\mathrm{OPT}^{(m)}} \\ &\leq \frac{\sum_{j \in J} w_j C_j^{(1)}}{\mathrm{OPT}^{(1)}} + 1 - \frac{1}{m} \end{aligned}$$

by $\mathrm{OPT}^{(m)} \geq \sum_{j \in J} w_j P_j$. Taking expectations completes the proof. □

Corollary 3.7. *Let ALG be any online list scheduling algorithm for*

$$P \mid P_j \sim F(\mu_j) \in \mathrm{NBUE}_{\mathrm{OPT}} \mid \sum_j w_j C_j.$$

If $\Pr[X_{j,k} = 1] \leq \alpha < 1$ *holds for all* $j \leq^{\pi} k$ *in all ALG lists* π, *then*

$$R(\mathrm{ALG}, \mathrm{NBUE}_{\mathrm{OPT}}) \leq \frac{1}{1-\alpha} + 1 - \frac{1}{m}.$$

Corollary 3.8. *The WSEPT algorithm for the stochastic scheduling problem* $P \mid P_j \sim \mathrm{Exp}(\mu_j^{-1}) \mid \sum_j w_j C_j$ *yields*

$$\mathbb{E}\left[\frac{\mathrm{WSEPT}}{\mathrm{OPT}}\right] \leq 3 - \frac{1}{m}.$$

Acknowledgement

The authors thank the anonymous referees for references and suggestions which helped improve the paper.

References

[1] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, Cambridge, 1998.

[2] E. C. Bruno, Jr., and R. Sethi. Scheduling independent tasks to reduce mean finishing time. *Communications of the ACM*, 17:382–387, 1974.

[3] J. Bruno, P. Downey, and G. N. Frederickson. Sequencing tasks with exponential service times to minimize the expected flow time or makespan. *Journal of the ACM*, 28(1):100–113, 1981.

[4] E. G. Coffman, Jr., and E. N. Gilbert. On the expected relative performance of list scheduling. *Operations Research*, 33(3):548–561, 1985.

[5] R. W. Conway, W. L. Maxwell, and L. W. Miller. *Theory of Scheduling*. Addison-Wesley, Reading, MA, 1967.

[6] W. L. Eastman, S. Even, and I. M. Isaacs. Bounds for the optimal scheduling of n jobs on m processors. *Management Science*, 11:268–279, 1964.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability—A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.

[8] I. Gertsbakh. *Statistical Reliability Theory*. Marcel Dekker, New York, 1989.

[9] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling theory: a survey. *Annals of Discrete Mathematics*, 5:287–326, 1979.

[10] L. A. Hall, A. S. Schulz, D. B. Shmoys, and J. Wein. Scheduling to minimize average completion time: off-line and on-line approximation algorithms. *Mathematics of Operations Research*, 22:513–544, 1997.

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

[12] T. Kawaguchi and S. Kyan. Worst case bound of an LRF schedule for the mean weighted flow-time problem. *SIAM Journal on Computing*, 15(4):1119–1129, 1986.

[13] E. Koutsoupias and C. Papadimitriou. Beyond competitive analysis. *Proceedings of the* 35*th Annual Symposium on Foundations of Computer Science* (*FOCS* '94), pages 394–400, 1994.

[14] B. J. Lageweg and J. K. Lenstra. Unpublished manuscript, 1977.

[15] R. H. Möhring, A. S. Schulz, and M. Uetz. Approximation in stochastic scheduling: the power of LP-based priority rules. *Journal of the ACM*, 46:924–942, 1999.
[16] C. Phillips, C. Stein, and J. Wein. Scheduling jobs that arrive over time. *Proceedings of the 4th Workshop on Algorithms and Data Structures* (*WADS* '95), pages 86–97. Volume 955 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, 1995.
[17] C. A. Phillips, C. Stein, and J. Wein. Minimizing average completion time in the presence of release dates. *Mathematical Programming*, 82:199–223, 1998.
[18] M. Pinedo. *Scheduling—Theory, Algorithms, and Systems*. Prentice–Hall, Englewood Cliffs, NJ, 1995.
[19] M. Pinedo and R. Weber. Inequalities and bounds in stochastic shop scheduling. *SIAM Journal on Applied Mathematics*, 44(4):867–879, 1984.
[20] C. M. Potts. An algorithm for the single machine sequencing problem with precedence constraints. *Mathematical Programming Study*, 13:78–87, 1980.
[21] M. H. Rothkopf. Scheduling with random service times. *Management Science*, 12:707–713, 1966.
[22] S. K. Sahni. Algorithms for scheduling independent tasks. *Journal of the ACM*, 23:116–127, 1976.
[23] M. Scharbrodt, T. Schickinger, and A. Steger. A new average case analysis for completion time scheduling. *Proceedings of the 34th Annual ACM Symposium on Theory of Computing* (*STOC* '02), pages 170–178, 2002. (Journal version accepted for publication by *Journal of the ACM*.)
[24] A. S. Schulz. Scheduling to minimize total weighted completion time: performance guarantees of LP-based heuristics and lower bounds. *Proceedings of the International Conference on Integer Programming and Combinatorial Optimization*, pages 301–315. Volume 1084 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, 1996.
[25] M. Skutella. Semidefinite relaxations for parallel machine scheduling. *Proceedings of the 39th Annual Symposium on Foundations of Computer Science* (*FOCS* '98), pages 472–481, 1998.
[26] M. Skutella and G. J. Woeginger. A PTAS for minimizing the total weighted completion time on identical parallel machines. *Mathematics of Operations Research*, 25:63–75, 2000.
[27] W. E. Smith. Various optimizers for single stage production. *Naval Research Logistics Quarterly*, 3:59–66, 1956.
[28] R. R. Weber, P. Varaiya, and J. Walrand. Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flowtime. *Journal of Applied Probability*, 23:841–847, 1986.
[29] G. Weiss and M. Pinedo. Scheduling tasks with exponential service times on non-identical processors to minimize various cost functions. *Journal of Applied Probability*, 17:187–202, 1980.

Received March 8, 2004, *and in final form July* 7, 2005. *Online publication November* 8, 2005.