Approximation algorithms for scheduling problems with a modified total weighted tardiness objective

Stavros G. Kolliopoulos [a,*], George Steiner [b]

[a] *Department of Informatics and Telecommunications, National and Kapodistrian University of Athens*

[b] *Management Science and Information Systems, McMaster University*

Abstract

We study the approximability of minimum total weighted tardiness with a modified objective which includes an additive constant. This ensures the existence of a positive lower bound for the minimum value. Moreover the new objective has a natural interpretation in Just-In-Time production systems.

Key words: Approximation Algorithms; Scheduling; Tardiness; Just-in-Time Scheduling.

1. Introduction

Minimizing the total weighted tardiness is one of the classical scheduling objectives studied by many researchers starting from the early days of scheduling theory [11]. We are given a set N of n jobs with the following characteristics. Job $j, 1 \leq j \leq n$, has to be processed for an integer time p_j on one of m ($m \geq 1$) machines, it has a due date d_j, and a positive weight w_j. For a given schedule of the jobs the *tardiness* T_j of job j is defined as $\max\{C_j - d_j, 0\}$, where C_j is the *completion time* of the job. The objective is to find the schedule which minimizes $\sum_{j=1}^{n} w_j T_j$. In the 3-field notation used in scheduling [11], the problem is denoted by $\alpha|\beta|\sum_j w_j T_j$, where α is the machine environment and β describes special job characteristics. Some possible values for α are 1 (single machine), P (identical parallel machines), Q (uniformly related machines) and R (unrelated machines). Rm stands for unrelated machines whose number is fixed. In the case of uniformly related machines, machine i has a speed $s_i \geq 1$. The processing of job j on machine i requires p_j/s_i time units. In the case of unrelated machines, job j takes p_{ij} units of processing time if assigned to machine i. Some possible values for β are r_j (denotes release dates, i.e., job j becomes available for processing at time $r_j > 0$), *prec* (denotes that the jobs are precedence-constrained), *pmtn* (the jobs can be preempted, i.e., interrupted and later restarted).

According to [7] $1|\ |\sum_j w_j T_j$ is an "NP-hard archetypal machine scheduling problem" whose exact solution appears very difficult even on very small inputs. We proceed to review briefly what is known on minimizing total weighted tardiness on a single machine.

Early on the problem was shown to be NP-hard in the ordinary sense [20] when the jobs have only two distinct due dates by a reduction from the knapsack problem. Much later even the case of a single common due date was shown NP-hard [30]. The problem was shown to be strongly NP-hard for an arbitrary number of due dates in [17]. Lawler and Moore [19] have presented a pseudopolynomial so-

* Corresponding author. Address: Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Panepistimiopolis Ilissia, Athens 157 84, Greece.

Email address: `steiner@mcmaster.ca` (George Steiner).

URL: `www.di.uoa.gr/~sgk` (Stavros G. Kolliopoulos).

lution for the case when all jobs have a single common due date. Very little is known about approximation algorithms. The only case that seems to be better understood is the usually easier case of *agreeable weights:* in that case $p_j < p_i$ implies $w_j \geq w_i$. Lawler gave a pseudopolynomial algorithm for the agreeable-weighted case [17]. A few years later he showed how to modify that algorithm to obtain an FPTAS for the case of unit weights [18]. Interestingly, the complexity of the unit weight problem, $1|\ |\sum_j T_j$, was open for many years until it was shown NP-hard [8]. Kolliopoulos and Steiner [15] gave pseudopolynomial algorithms for the case of $1|\ |\sum_j w_j T_j$ when there is only a fixed number of different due dates. They have also developed an FPTAS if, in addition, the weights w_j are bounded by a polynomial function of n. Cheng, Ng, Yuan and Liu [4] have recently shown that the schedule which minimizes $\max_j w_j T_j$ yields an $(n-1)$-approximation for $1|\ |\sum_j w_j T_j$. To our knowledge this is the only nontrivial known approximation guarantee for a version of the problem with general input data. We are not aware of any approximation guarantees for multimachine weighted tardiness scheduling problems.

Minimizing total tardiness becomes provably hard to approximate with an extension to $1|\ |\sum_j T_j$ as mild as that of the jobs having nontrivial release dates. The *flow time* F_j of a job j is defined as $C_j - r_j$. There is a straighforward approximation-preserving reduction from minimizing total flow time on a single machine to $1|r_j|\sum_j T_j$. Therefore from the hardness result of [14] it is NP-hard to obtain an $O(n^{1/2-\varepsilon})$-approximation for the latter problem for any $\varepsilon > 0$.

The proposed model. In this paper we attempt to tackle this apparently very difficult class of problems from a different perspective. We present a large number of approximation results for scheduling problems of the form $\alpha|\beta|\sum_j w_j(T_j + d_j)$. It is clear that for optimization purposes this modified objective is equivalent to minimizing the total weighted tardiness as it adds the schedule-independent constant $\sum_j w_j d_j$ to the objective. Part of the difficulty of finding good multiplicative approximations for $\sum_j w_j T_j$ seems to be that the optimum can be zero (one can determine whether this is the case for $1|\ |\sum_j w_j T_j$ by scheduling the jobs in the Earliest Due Date (EDD) order) or in general very small compared to the job processing times. This type of irregularity arises for other objectives too, for example for the maximum lateness $L_{\max}$. For job j, the *lateness* L_j is defined as $C_j - d_j$. The usual way to deal with this difficulty is to use a formulation which adds a positive constant to the objective. In the lateness setting the added constant transforms the lateness of job j from $C_j - d_j$ to $C_j + q_j$ where $q_j > 0$, is the so-called delivery time of the job. All known approximation results for minimizing "lateness" are for this modified objective. The conceptual starting point of the described transformation is to let the due dates be nonpositive and then interpret $-d_j$ as expressing the nonnegative delivery time. The resulting metric, which depends on $C_j + q_j$, turns out to be interesting in its own right but it actually eliminates the due dates and the lateness from the problem statement. See the survey [12] for an extensive discussion of this type of transformation.

Our modified objective maintains information about the tardy jobs and has a natural interpetation of its own. In particular it has an interesting application in Just-in-Time (JIT) production systems. An example of such a system is a manufacturer supplying parts for the auto industry. As the name suggests, it is desirable in a JIT system for all jobs (e.g., car parts) to be completed as close to their due date as possible. This usually results in substantial reduction of work-in-process inventory and thus inventory carrying costs, which are proportional to the length of time each job spends in the system. The customers (e.g. auto industry for the parts) require JIT delivery of the jobs. It is then reasonable to assume that early jobs get delivered, i.e., leave the system, only when they are due and tardy jobs are delivered as soon as they are completed. It is easy to see that our modified objective function $\sum_j w_j(T_j + d_j)$ corresponds to this as it equals the sum of weighted completion times of the tardy jobs plus the sum of the weighted due dates of the early jobs. Then the total work-in-process inventory carrying cost for the manufacturer is exactly $\sum_j w_j(T_j + d_j)$, where w_j represents the cost of holding job j in inventory for one unit of time.

Kovalyov and Werner [16] have studied the approximability of the unit-weight case on parallel machines with common due date, i.e., $Pm|d_j = d|\sum_j T_j$. Using the fact that it is NP-complete to decide whether there is a schedule with zero tardiness, they prove that, unless $P = NP$, there is no polynomial-time ρ-approximation algorithm for $Pm|d_j = d|\sum_j T_j$ with $\rho < \infty$. This shows that solutions with zero value for the objective function must be avoided to have any hope for a constant factor approximation. We can do this by adding an appropriate quantity $b > 0$ to the objective function. They also show, however, that unless $P = NP$, there is no polynomial-time $(\rho+1)-$approximation algorithm for $Pm|d_j = d|\sum_j T_j + b$ with $\rho < 1/b$. This also implies that there is no FPTAS for the problem if b is bounded by a polynomial in the input length of the instance unless $P = NP$. They proceed to construct an FPTAS for $Pm|d_j = d|\sum_j T_j + d$. Our objective function extends this modification to the weighted case with different due dates.

Approximation ratios. Our results are based on exploiting the close relationship between the $\sum_j w_j C_j$ and $\sum_j w_j(T_j + d_j)$ objectives in a large number of scheduling environments. The approximability of minimizing total weighted completion time is much better understood, cf. the survey [3]. We prove in Section 2 that approximating $\sum_j w_j(T_j + d_j)$ reduces to approximating $\sum_j w_j C_j$ in the sense that any ρ-approximation algorithm for minimizing $\sum_j w_j C_j$ is a $(\rho + 1)$-approximation algorithm for minimizing $\sum_j w_j(T_j + d_j)$. We also show that it is possible to further improve upon these guarantees in cases where LP-based algorithms with certain characteristics are available for approximating the corresponding $\alpha|\beta|\sum_j w_j C_j$ problem. We propose a family of linear relaxations for the modified tardiness function and use it to prove that, in the cases mentioned, a schedule with a ρ-approximation ratio for $\alpha|\beta|\sum_j w_j C_j$ is also a schedule with the *same* approximation ratio for the corresponding $\alpha|\beta|\sum_j w_j(T_j+d_j)$ problem. Our final contribution in Section 3 is an FPTAS for the single-machine case where all the jobs have a common due date D, i.e., for $1|d_j = D|\sum_j w_j(T_j + d_j)$. This FPTAS works without any restricting assumptions about the weights.

2. Reduction to total weighted completion time

In this section we show how to reduce the problem of finding an approximate solution to minimizing $\sum_j w_j(T_j + d_j)$ to the problem of finding an approximate solution to $\sum_j w_j C_j$. Using the 3-field scheduling notation we examine problems belonging to the family $\alpha|\beta|\sum_j w_j(T_j + d_j)$.

For any schedule σ let $C_j(\sigma), T_j(\sigma)$ denote the completion time and tardiness of job j in schedule σ. Note also that $T_j + d_j = \max\{C_j, d_j\}$. For an arbitrary (preemptive or nonpreemptive) schedule σ it holds that $\sum_j w_j C_j(\sigma) \le \sum_j w_j \max\{C_j(\sigma), d_j\} \le \sum_j w_j C_j(\sigma) + \sum_j w_j d_j$.

It follows from the first inequality that the minimal total weighted completion time $\mathrm{OPT}_{\sum_j w_j C_j}$ is a lower bound on the value $\mathrm{OPT}_{\sum_j w_j \max\{C_j,d_j\}}$ of an optimal solution for the modified objective function. Another trivial lower bound for this optimum is the value $\sum_j w_j d_j$. Therefore any schedule σ^ρ with $\sum_j w_j C_j(\sigma^\rho) \le \rho \mathrm{OPT}_{\sum_j w_j C_j}$ fulfills

$$\sum_j w_j \max\{C_j(\sigma^\rho), d_j\} \le \rho \mathrm{OPT}_{\sum_j w_j C_j} + \sum w_j d_j \le (\rho+1)\mathrm{OPT}_{\sum_j w_j \max\{C_j,d_j\}}.$$

Thus we have proved the following.

Theorem 1 *Consider a member $\alpha_0|\beta_0|\sum_j w_j C_j$ of the family of scheduling problems $\alpha|\beta|\sum_j w_j C_j$ for which there is a ρ-approximation algorithm. Then the same algorithm achieves a $(\rho+1)$-approximation for the problem $\alpha_0|\beta_0|\sum_j w_j(T_j + d_j)$.*

We summarize the most important consequences of Theorem 1 in Table 1. We omit from the table bounds that are improved later on with an LP-based approach.

An extension of Theorem 1 results as follows. One could have a stochastic input where the vector $\mathbf{P}$ of processing times of the jobs is a vector of random variables from known distributions. In that case the solution of a problem is no longer a simple schedule but a scheduling policy Π [22] which yields a feasible schedule for each realization $\mathbf{p}$ of the processing time vector. Accordingly the performance of a policy under the total weighted completion time objective is a random variable $Z^\Pi(\mathbf{P})$ and an optimal policy Π^* is one that minimizes the expectation $E[Z^\Pi(\mathbf{P})]$. A policy Π is a *ρ-approximation* if

$$E[Z^\Pi(\mathbf{P})] \le \rho E[Z^{\Pi^*}(\mathbf{P})].$$

See e.g, [22,29] for more details. Under this new definition of OPT for the stochastic case and taking expectations where needed in the relations above, we can conclude that the stochastic analogue of Theorem 1 holds as well. A number of constant-factor approximation results of this type are shown in [23,29] for $P|r_j, prec|E[\sum_j w_j C_j]$ and its special cases. Under the same probabilistic assumptions these results translate to constant-factor (increased by one) approximations for $P|r_j, prec|E[\sum_j w_j(T_j + d_j)]$.

It is possible to improve upon the guarantee of Theorem 1 and bring down the approximation ratio

Problem	ratio for $\sum_j w_j(T_j+d_j)$	reference for $\sum_j w_j C_j$
$P\|r_j,pmtn\|\sum_j w_j(T_j+d_j)$	$2+\varepsilon$	[1]
$P\|r_j\|\sum_j w_j(T_j+d_j)$		
$Rm\|r_j,pmtn\|\sum_j w_j(T_j+d_j)$		
$Rm\|r_j\|\sum_j w_j(T_j+d_j)$		
$Q\|pmtn\|\sum_j (T_j+d_j)$	2	[10]
$Q\|r_j\|\sum_j w_j(T_j+d_j)$	$2+\varepsilon$	[2]

Table 1
Approximation ratios $\rho+1$ obtained by Theorem 1 for various instantiations of $\alpha|\beta|\sum_j w_j(T_j+d_j)$. The references give the sources for the corresponding ρ approximation ratio achieved for $\alpha|\beta|\sum_j w_jC_j$. We omit bounds that are improved by Theorem 3.

by 1 in cases where LP-based algorithms with certain characteristics are available for approximating the total weighted completion time.

The *earliness* of a job j in a schedule σ is defined as $E_j(\sigma) := \max\{d_j - C_j(\sigma), 0\}$. We will use mathematical programming formulations with variables T_j, E_j, C_j to denote respectively the tardiness, earliness, and completion time of job j, $j=1,\ldots,n$. We propose a family of mathematical programs which is parameterized based on a set of constraints $\mathcal{C}(C)$. In principle the constraints in $\mathcal{C}(C)$ are general convex. A set of constraints $\mathcal{C}(C)$ is a *a valid set of completion time constraints* if the completion times C_j $(j=1,2,...,n)$ must satisfy the constraints $\mathcal{C}(C)$ for *every* feasible schedule.

As a concrete example for $\mathcal{C}(C)$, consider the problem $1|prec|\sum_j w_j(T_j+d_j)$, i.e., scheduling precedence-constrained jobs on a single machine. A valid set of completion time constraints, which was introduced in [26], is the following:

$$C_k \geq C_j + p_k \quad \text{for each pair } j,k \text{ s.t.} j \prec k$$

$$\sum_{j\in S} p_jC_j \geq \frac{1}{2}\left(p^2(S)+p(S)^2\right) \quad \forall S\subseteq N$$

Here $p(S)=\sum_{j\in S}p_j$, $p^2(S)$ denotes $\sum_{j\in S}p_j^2$ and $j \prec k$ represents the precedence constraint that job j has to be finished before job k can start processing. Queyranne has shown that a separation oracle exists for the exponentially large set of constraints and hence one can optimize over them in polynomial time. Various other valid sets have also been given in the literature. See for example the ones based on linear-ordering variables [25,5,21] and the time-indexed formulation in [9]. Note that the first two of these use constraint sets of polynomial size.

Let $\mathcal{C}(C)$ be a valid set of completion time constraints for problem $\alpha|\beta|\sum_j w_j(T_j+d_j)$. We emphasize that the T_j or E_j variables do not have to appear in any constraint in $\mathcal{C}(C)$. The only variables involved may be completion time variables and possibly other auxiliary ones. In fact, in all the formulations we employ the tardiness and earliness variables do not appear in $\mathcal{C}(C)$. We propose the following family of linear programs, denoted $FP(\mathcal{C})$:

$$\text{minimize} \sum_{j=1}^{n} w_j(T_j+d_j) \tag{1}$$

$$T_j = C_j - d_j + E_j \quad j=1,\ldots,n \tag{2}$$

$$\mathcal{C}(C) \tag{3}$$

$$T_j, E_j, C_j \geq 0 \quad j=1,\ldots,n \tag{4}$$

Since every job is either early or tardy in any feasible schedule, i.e., at most one of the two variables T_j and E_j can be positive, it is easy to see that the T_j and E_j values must satisfy (2) for any feasible schedule. Of course equations (2) do allow solutions in which both T_j and E_j are positive, thus $FP(\mathcal{C})$ is normally not an exact formulation for the problem $\alpha|\beta|\sum_j w_j(T_j+d_j)$, it is only a linear programming relaxation.

Our results are based on the following algorithm schema for the generic problem $\alpha|\beta|\sum_j w_j(T_j+d_j)$. The values of the completion time variables returned by an optimal solution of $FP(\mathcal{C})$ may not be integer and we refer to these values as the *fractional completion times.* Our schema assumes the existence of a subroutine $\mathcal{A}(\alpha,\beta)$ which finds a feasible schedule σ that comes with a *job-by-job approximation guarantee* for the completion times, i.e., if $\overline{C}_j$ is the fractional completion time and $C_j(\sigma)$ is the completion time in σ for job j, then we have $C_j(\sigma) \leq \rho\overline{C}_j$, $j=1,\ldots,n$, for some $\rho \geq 1$.

Algorithm Schema($\mathcal{C}$)
1. Compute an optimal solution to $FP(\mathcal{C})$. Let

$\overline{T}_j, \overline{E}_j\ \overline{C}_j$, be the resulting values of the variables, $j = 1, \ldots, n$.

2. By invoking an appropriate algorithm $\mathcal{A}(\alpha, \beta)$, compute a feasible schedule σ for $\alpha|\beta|\sum_j w_j C_j$ in which $C_j \leq \rho \overline{C}_j$, $j = 1, \ldots, n$, for some $\rho \geq 1$.

3. Output the schedule σ.

The analysis of our various algorithms hinges on the following crucial lemma.

Lemma 2 *If the algorithm $\mathcal{A}(\alpha, \beta)$ assumed in Step 2 of the* Algorithm Schema *exists, the output schedule σ achieves a ρ-approximation for problem $\alpha|\beta|\sum_j w_j(T_j + d_j)$.*

PROOF. The schedule σ is feasible for the machine environment α and job characteristics β by construction. Denote by $T_j(\sigma), E_j(\sigma)$ and $C_j(\sigma)$ the tardiness, earliness and completion time of job j in schedule σ. Then for $j = 1, \ldots, n$,

$$T_j(\sigma) = C_j(\sigma) - d_j + E_j(\sigma) \leq \rho \overline{C}_j - d_j + E_j(\sigma). \tag{5}$$

If job j is not tardy, $T_j(\sigma) = 0$ and the contribution of job j to the objective is $w_j d_j$. If job j is tardy, the contribution of j to the objective is $w_j(T_j(\sigma) + d_j)$ which by (5) is at most

$$w_j(T_j(\sigma) + d_j) \leq w_j(\rho \overline{C}_j + E_j(\sigma)) = \rho w_j \overline{C}_j.$$

But from (2)

$$w_j(\overline{T}_j + d_j) = w_j(\overline{C}_j + \overline{E}_j),$$

hence $w_j(T_j(\sigma) + d_j)$ equals

$$w_j C_j(\sigma) \leq \rho w_j \overline{C}_j \leq \rho w_j(\overline{T}_j + d_j)$$

and the lemma is shown.

□

The analysis of Lemma 2 holds also in the case where the algorithm $\mathcal{A}(\alpha, \beta)$ returns a preemptive schedule, in which case the schedule σ output by the Algorithm Schema will be preemptive as well. The following theorem has been shown.

Theorem 3 *If for a problem $\alpha|\beta|\sum_j w_j(T_j + d_j)$ (i) a valid set of completion time constraints exists (ii) $FP(\mathcal{C})$, where $\mathcal{C}(C)$ is replaced by a specific valid set, can be solved in polynomial time and (iii) a ρ-approximation algorithm for $\sum_j w_j C_j$ with the job-by-job guarantee exists, then there is ρ-approximation algorithm for $\sum_j w_j(T_j + d_j)$.*

The main problems for which our requirements are satisfied are shown in Table 2.

We remark that each ratio ρ in Table 2 implies the same upper bound ρ on the integrality gap of the corresponding linear relaxation, i.e., of the corresponding member of the family $FP(\mathcal{C})$ of linear relaxations. Furthermore, $\alpha|\beta|\sum_j w_j C_j$ is a special case of $\alpha|\beta|\sum_j w_j(T_j + d_j)$ with $d_j = 0$ for $j = 1, ..., n$, and all our inequalities reduce to the ones describing the associated $\alpha|\beta|\sum_j w_j C_j$ problem in this case. Therefore, if a ratio is known to be tight for a relaxation of $\alpha|\beta|\sum_j w_j C_j$, then it is also tight for the corresponding $\alpha|\beta|\sum_j w_j(T_j + d_j)$ problem.

3. An FPTAS for the common due date case

In this section we will present an FPTAS for the problem $1|d_j = D|\sum w_j(T_j + d_j)$, i.e., scheduling on a single machine when all the jobs have the same due date D. Even this special case is NP-hard as shown by Yuan [30]. Lawler and Moore [19] have presented an $O(n^2 D)$ pseudopolynomial algorithm for this problem. We show how to transform this pseudopolynomial algorithm to an FPTAS.

Similarly to [18] and [15], we are going to scale and round down the processing times by a constant K, which is to be determined later. Unlike [18] and [15], we will not only scale down the due dates by the same constant K but we will also round them to an integer value.

Accordingly, let us define $\overline{d}_j := \lceil d_j/K \rceil$ and $\overline{p}_j := \lfloor p_j/K \rfloor$ for $j = 1, 2, ..., n$. Assume that we apply the Lawler-Moore dynamic programming algorithm to this scaled down problem and let σ_A be the optimal sequence found by the algorithm. Let σ^* be the sequence minimizing $\sum_{j=1}^n w_j T_j$ and denote by $T(\sigma^*)$ this optimum value. Let $\overline{T}_{\sigma_A(j)}$ be the tardiness of the jth job in the sequence σ_A with the scaled down data (i.e., $\overline{p}_j$ and $\overline{d}_j$) and let $T_{\sigma_A(j)}$ be the tardiness of the same job in σ_A with the original data. In addition, define $T_{\sigma_A} := \sum_{j=1}^n w_{\sigma_A(j)} T_{\sigma_A(j)}$. Then we clearly have $\overline{T}_{\sigma_A(j)} \leq T_{\sigma_A(j)}/K$ for $j = 1, 2, ..., n$. Furthermore, $\overline{T}_{\sigma_A} := \sum_{j=1}^n w_{\sigma_A(j)} \overline{T}_{\sigma_A(j)} \leq T(\sigma^*)/K$ since σ_A is optimal for the scaled down data. Let T'_{σ_A} denote the total weighted tardiness of the sequence σ_A when we use processing times $p'_j := K\overline{p}_j$ for each job j and the original due dates d_j. Note that $p'_j = K\overline{p}_j \leq p_j \leq K(\overline{p}_j + 1)$. We can write

$$K\overline{T}_{\sigma_A} \leq T(\sigma^*) \leq T_{\sigma_A}$$

which is less than

Problem	ratio for $\sum_j w_j(T_j + d_j)$	reference for $\sum_j w_j C_j$
$1\|prec\|\sum_j w_j(T_j + d_j)$	$\rho = 2$	[13]
$1\|r_j, prec, pmtn\|\sum_j w_j(T_j + d_j)$	$\rho = 2$	[13]
$1\|r_j, prec\|\sum_j w_j(T_j + d_j)$	$\rho = e + \varepsilon$	[27]
$P\|r_j, prec, pmtn\|\sum_j w_j(T_j + d_j)$	$\rho = 3$	[13]
$P\|r_j, prec\|\sum_j w_j(T_j + d_j)$	$\rho = 4$	[24]
$Q\|r_j, prec, pmtn\|\sum_j w_j(T_j + d_j)$	$\rho = O(\log m)$	[6]
$Q\|r_j, prec\|\sum_j w_j(T_j + d_j)$		
$R\| \|\sum_j w_j(T_j + d_j)$	1.5	[28]
$R\|r_j\|\sum_j w_j(T_j + d_j)$	2	[28]
$R\|pmtn\|\sum_j w_j(T_j + d_j)$	2	[28]
$R\|r_j, pmtn\|\sum_j w_j(T_j + d_j)$	3	[28]

Table 2
Approximation ratios achieved by Theorem 3 for various instantiations of $\alpha|\beta|\sum_j w_j(T_j + d_j)$. The references give the sources for the corresponding $\mathcal{A}(\alpha, \beta)$ subroutine required by the ALGORITHM SCHEMA.

$$\sum_{j=1}^{n} w_{\sigma_A(j)} \max\{K \sum_{i=1}^{j} (\overline{p}_{\sigma_A(i)} + 1) - d_{\sigma_A(j)}, 0\} \leq T'_{\sigma_A} + Kn \sum_{j=1}^{n} w_j.$$

Furthermore,

$$K\overline{T}_{\sigma_A} = K \sum_{j=1}^{n} w_{\sigma_A(j)} \max\{\sum_{i=1}^{j} \overline{p}_{\sigma_A(i)} - \overline{d}_{\sigma_A(j)}, 0\}$$

which is at least

$$K \sum_{j=1}^{n} w_{\sigma_A(j)} \max\{\sum_{i=1}^{j} \overline{p}_{\sigma_A(i)} - (\frac{d_{\sigma_A(j)}}{K} + 1), 0\}.$$

The latter quantity can be lower bounded by $\sum_{j=1}^{n} w_{\sigma_A(j)} \max\{\sum_{i=1}^{j} K\overline{p}_{\sigma_A(i)} - d_{\sigma_A(j)}, 0\} - K\sum_{j=1}^{n} w_{\sigma_A(j)} = T'_{\sigma_A} - K\sum_{j=1}^{n} w_j$.

Combining the above we obtain $T'_{\sigma_A} - K\sum_{j=1}^{n} w_j \leq T(\sigma^*) \leq T_{\sigma_A} \leq T'_{\sigma_A} + Kn\sum_{j=1}^{n} w_j$, which implies

$$T_{\sigma_A} - T(\sigma^*) \leq K(n+1) \sum_{j=1}^{n} w_j \tag{6}$$

Choose $K = \varepsilon D/(n+1)$. By (6) the error is at most $\varepsilon \sum_j w_j D = \varepsilon \sum_j w_j d_j \leq \varepsilon OPT$. The running time of the algorithm that computes σ_A on the scaled input is $O(n^2(D/K)) = O(n^3/\varepsilon)$. We have proved the following theorem.

Theorem 4 *There is an FPTAS for the problem $1|d_j = D|\sum_j w_j(T_j + d_j)$, i.e., minimizing $\sum_j w_j(T_j + d_j)$ on a single machine when all the jobs have a common due date.*

Acknowledgements

We thank an anonymous referee for concrete suggestions on how to shorten the proofs. Stavros Kolliopoulos was partially supported by the European Social Fund and National Resources - (EPEAEK-II) ARCHIMIDIS II. George Steiner was partially supported by NSERC Grant OG0001798.

References

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

[2] C. Chekuri, S. Khanna, A PTAS for minimizing weighted completion time on uniformly related machines, in: Proceedings of the 28th ICALP, 2001.

[3] C. Chekuri, S. Khanna, Approximation algorithms for minimizing the weighted sum of completion times, in: J. Y.-T. Leung (ed.), Handbook of Scheduling: Algorithms, Models, and Performance Analysis, Chapman & Hall/CRC, 2004.

[4] T. C. E. Cheng, C. T. Ng, J. J. Yuan, Z. H. Liu, Single machine scheduling to minimize total weighted tardiness, European Journal of Operational Research 165 (2005) 423–443.

[5] F. A. Chudak, D. S. Hochbaum, A half-integral linear programming relaxation for scheduling precedence-

constrained jobs on a single machine, Operations Research Letters 25 (1999) 199–204.
[6] F. A. Chudak, D. B. Shmoys, Approximation algorithms for precedence-constrained scheduling problems on parallel machines that run at different speeds, Journal of Algorithms 30 (1999) 323–343.
[7] R. K. Congram, C. N. Potts, S. L. van de Velde, An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem, INFORMS Journal on Computing 14 (1) (2002) 52–67.
[8] J. Du, J. Y.-T. Leung, Minimizing total tardiness on one machine is NP-hard, Mathematics of Operations Research 15 (1990) 483–495.
[9] M. Dyer, L. Wolsey, Formulating the single machine sequencing problem with release dates as a mixed integer program, Discrete Applied Mathematics 26 (1990) 255–270.
[10] T. Gonzalez, Optimal mean finish time preemptive schedules, Tech. Rep. Tech. Rep. 220, Comp. Sci. Dept., Pennsylvania State University (1977).
[11] R. L. Graham, E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, Annals of Discrete Mathematics 5 (1979) 287–326.
[12] L. A. Hall, Approximation algorithms for scheduling, in: D. S. Hochbaum (ed.), Approximation Algorithms for NP-hard problems, PWS, 1997, pp. 1–45.
[13] L. A. Hall, A. Schulz, D. B. Shmoys, J. Wein, Scheduling to minimize average completion time: Off-line and on-line approximation algorithms, Mathematics of Operations Research 22 (1997) 513–544.
[14] H. Kellerer, T. Tautenhahn, G. J. Woeginger, Approximability and nonapproximability results for minimizing total flow time on a single machine, in: Proceedings of the 28th Annual ACM Symposium on Theory of Computing, 1996.
[15] S. G. Kolliopoulos, G. Steiner, Approximation algorithms for minimizing the total weighted tardiness on a single machine, Theoretical Computer Science 355 (2006) 261–273.
[16] M. Y. Kovalyov, F. Werner, Approximation schemes for scheduling jobs with common due date on parallel machines to minimize total tardiness, Journal of Heuristics 8 (2002) 415–428.
[17] E. L. Lawler, A "pseudopolynomial" algorithm for sequencing jobs to minimize total tardiness, Annals of Discrete Mathematics 1 (1977) 331–342.
[18] E. L. Lawler, A fully polynomial approximation scheme for the total tardiness problem, Operations Research Letters 1 (1982) 207–208.
[19] E. L. Lawler, J. M. Moore, A functional equation and its application to resource allocation and sequencing problems, Management Science 16 (1969) 77–84.
[20] J. Lenstra, A. R. Kan, P. Brucker, Complexity of machine scheduling problems, Annals of Discrete Mathematics 1 (1977) 343–362.
[21] F. Margot, M. Queyranne, Y. Wang, Decompositions, network flows and a precedence constrained single-machine scheduling problem, Operations Research 51 (2003) 981–992.
[22] R. H. Möhring, F. J. Radermacher, G. Weiss, Stochastic scheduling problems I: general strategies, ZOR - Zeitschrift für Operations Research 28 (1984) 193–260.
[23] R. H. Möhring, A. S. Schulz, M. Uetz, Approximation in stochastic scheduling: the power of LP-based priorities, Journal of the ACM 46 (1999) 924–942.
[24] A. Munier, M. Queyranne, A. Schulz, Approximation bounds for a general class of precedence constrained parallel machine scheduling problems, in: Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization, 1998.
[25] C. N. Potts, An algorithm for the single machine sequencing problem with precedence constraints, Mathematical Programming Study 13 (1980) 78–87.
[26] M. Queyranne, Structure of a simple scheduling polyhedron, Mathematical Programming 58 (1993) 263–285.
[27] A. Schulz, M. Skutella, Random-based scheduling: New approximations and lp lower bounds, in: Proc. 1st Int. Symp. on Randomization and Approximation Techniques in Computer Science (RANDOM), 1997.
[28] M. Skutella, Convex quadratic and semidefinite programming relaxations in scheduling, Journal of the ACM 48 (2001) 206–242.
[29] M. Skutella, M. Uetz, Stochastic machine scheduling with precedence constraints, SIAM J. Comput. 34 (2005) 788–802.
[30] J. Yuan, The NP-hardness of the single machine common due date weighted tardiness problem, Systems Science and Mathematical Sciences 5 (1992) 328–333.