Approximation algorithms for minimizing the total weighted earliness on machines scheduling

Hussam A. Mohammed
Mathematical Department
College of Education
University of Kerbala
E-mail: hussam_altaai@yahoo.com

Abstract :
Minimizing the total weighted earliness is not well-studied objective in scheduling theory, which is very little understood from the point of view of approximation algorithms. In an attempt to make progress, we study the approximability of minimum total weighted earliness with a modified objective which includes an additive the total weighted completion time. 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.
Keyword: Scheduling, Weighted Earliness, Approximation

المستخلص :
تقليل مجموع أهمية التبكير لم يدرس جيداً في نظرية الجدولة وذلك لأن عدد قليل من الباحثين تطرقوا إلى هذه المسألة من وجهة نظر خوارزميات التقريب. في محاولة لإحراز تقدم, ندرس التقريب لتقليل مجموع أهمية التبكير مع دالة هدف معدلة والذي يتضمن بإضافة مجموع أهمية زمن أتمام العمل. هذا يضمن وجود حد أدنى إيجابي لأقل قيمة. علاوة على ذلك دالة الهدف الجديدة لها تفسير طبيعي في أنظمة الإنتاج في وقت التمام.

1 Introduction

Minimizing the total weighted earliness is one of the classical scheduling objectives studied by few researchers because this problem is equivalent to tardiness penalties (i.e., $\sum_j w_j E_j \approx \sum_j w_j T_j$). It is well-known that $\sum_j w_j T_j$ is *NP*-hard problem and most researches have studies the tardiness penalties, for more details (see [6], [15]). We are given a set N of n jobs with the following characteristics. Job j, $1 \le j \le n$, has to be processed for an integer time p_j on one of m $(m \ge 1)$ machines, it has a due date d_j, and a positive weight w_j. For a given schedule of the jobs the earliness $E_j$of job j is defined as $max\{d_j - C_j\}$, where C_j is the completion time of the job. The objective is to find the schedule which minimizes $\sum_{j=1}^{n} w_j E_j$. In the 3-field notation used in scheduling [18], the problem is denoted by $\alpha|\beta|\sum_j w_j E_j$, 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 $s_i \ge 1$. 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). We proceed to review briefly what is known on minimizing total weighted earliness on a single machine. Early on the problem was shown to be *NP*-hard in the ordinary sense by Lenstra, Rinnooy Kan and Brucker [27] when the jobs have only two distinct due dates by a reduction from the knapsack problem. Rios-Solis and Sourd [34] solved the parallel machine scheduling problem where jobs have different earliness-tardiness penalties and a restrictive common due date and defined an exponential size neighborhood for this problem and proved that finding the local minimum in it is an *NP*-hard problem and they proposed a pseudo-polynomial algorithm that finds the best solution of the exponential neighborhood. Chhajed [12] introduced a problem where *n* jobs must be assigned one of two due dates, which were given at equal interval. Jobs tardiness was not allowed. The due date cost was considered in addition to the earliness cost. In Asano and Ohta [5], not only the due date of each job was known, but also the ready time was prespecified. They proposed an optimization algorithm using dominance relation for the scheduling problem. Their algorithm was based on the branch and bound approach. Lawler and Moore [26] have presented a pseudopolynomial solution for the case when all jobs have a single common due date. Very little is known about approximation algorithms. Ahmadi and Bagchi [2] studied the scheduling problem in which the objective was to minimize the total earliness cost subject to meeting due dates. They showed that the problem was *NP*-complete and that the problem was equivalent to the problem of minimizing the total waiting time of all jobs subject to release times and common due date. They developed a branch and bound method to solve the problem. This later problem was first study by Ahmadi and Bagchi [3]. They developed a branch and bound method based on unequal weights to solve the problem Chand and Schneeberger [8] also studied the problem of earliness cost. They studied two types of problems, which they referred to as the Weighted Earliness Problem (*WE-Problem*), the Constrained Weighted Earliness Problem (*CWE-Problem*). In their WE-problem, machine idle time can be inserted, but idle time is not permitted in CWE-problem. They showed that both problems were *NP*-hard, and that optimal solution can be obtained by a polynomial time algorithm only in some specified situations. In their paper, they developed a dynamic programming based approach for solving the WE-problem. They also developed two heuristic algorithms for solving the CWE-problem and WE-problem, respectively. The heuristics were a modification of Smith heuristic [37]. 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 multi-machine weighted earliness scheduling problems. Ahmed [4] produced the earliness penalties on a single machine subject to no tardy jobs $1|C_j \le d_j|\sum_j w_j E_j$ and discussed the relationship between earliness and tardiness and proved a very good result, that the Earliest Due Date (EDD) rule with $E_j \le p_j$ is optimal for $1|C_j \le d_j|\sum_j E_j$ problem. Minimizing total earliness becomes provably hard to approximate with an extension to $1||\sum_j E_j$ 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 straightforward approximation-preserving reduction from minimizing total flow time on a single machine to $1|r_j|\sum_j E_j$. Therefore from the hardness result of Kellerer, Tautenhahn and Woeginger [22] 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 $\alpha|\beta|\sum_j w_j(E_j + C_j)$. It is clear that for optimization purposes this modified objective is equivalent to minimizing the total weighted earliness as it adds the completion time $\sum_j w_j C_j$ to the objective. Part of the difficulty of finding good multiplicative approximations for $\sum_j w_j E_j$ seems to be that the optimum can be zero (one can determine whether this is the case for $1||\sum_j w_j E_j$ by scheduling the jobs in the longest weighted processing time (LWPT) rule) or in general very big compared to the job processing times. This type of irregularity arises for other objectives too, for example for the maximum earliness E_{max}. For job j, the *earliness* E_j is defined as $d_j - C_j$. The usual way to deal with this difficulty is to use a formulation which adds a positive constant to the objective. In the earliness setting the added constant transforms the earliness of job *j* from $d_j - C_j$ to $C_j + q_j$ where $q_j > 0$, is the so-called delivery time of the job. All known approximation results for minimizing "earliness" 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 by Hall [19] for an extensive discussion of this type of transformation. Our modified objective maintains information about the early jobs and has a natural interpretation 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 tardy jobs get delivered, i.e., leave the system, only when they are completion and early jobs are delivered as soon as they are completed. It is easy to see that our modified objective function $\sum_j w_j(E_j + C_j)$ corresponds to this as it equals the sum of weighted completion times of the early jobs plus the sum of the weighted due dates of the early jobs (cf. Equation (1) below). Then the total work-in-process inventory carrying cost for the manufacturer is exactly $\sum_j w_j(E_j + C_j)$, where w_j represents the cost of holding job j in inventory for one unit of time.

The modified objective has a "smoothing" effect on the objective values, namely if the C_j-s are small the E_j-s are expected to be larger and if the C_j-s are large the E_j-s are expected to be smaller. Observe also that an *O*(1)-approximation for the modified objective yields a schedule whose total weighted earliness is at most a constant time the optimal total weighted earliness plus an additive factor which is a constant depending on $\sum_j w_j C_j$.

Approximation ratios. Our results are based on exploiting the close relationship between the $\sum_j w_j C_j$ and $\sum_j w_j(E_j + C_j)$ objectives in a large number of scheduling environments. The approximability of minimizing total weighted completion time is much better understood, cf. the survey of Chekuri and Khanna [10]. We prove that approximating $\sum_j w_j(E_j + C_j)$ reduces to appr-oximating $\sum_j w_j C_j$ in the sense that any ρ-approximation algorithm for minimizing $\sum_j w_j C_j$ is a $(\rho + 1)$-approximation algorithm for minimizing $\sum_j w_j(E_j + C_j)$. 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 earliness 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 $\alpha|\beta|\sum_j w_j(E_j + C_j)$ problem. Our final contri-bution is an FPTAS for the single-machine case where all the jobs have a common due date D, i.e., for $1|d_j = D|\sum_j w_j(E_j + C_j)$. This FPTAS works without any restricting assumptions about the weights.

The paper is organized as follows. In the next section we prove a meta theorem stating that a ρ-approximate schedule for $\alpha|\beta|\sum_j w_j C_j$ achieves a (ρ+1)-approximation for $\alpha|\beta|\sum_j w_j(E_j + C_j)$. Some of the immediate consequences of this result are the first constant-ratio approximation algorithms for a large number of modified total weighted earliness scheduling problems. These are summarized in Table 1. Section 3 contains another meta theorem which proves that an LP-based ρ-approximation algorithm also yields a ρ-approximation for our modified total weighted earliness objective under certain additional conditions. Some of the consequences of this are summarized in Table 2. The last section contains the FPTAS for $1|d_j = D|\sum_j w_j(E_j + C_j)$.

2 A general reduction

In this section we show how to reduce the problem of finding an approximate solution to minimizing $\sum_j w_j(E_j + C_j)$ to the problem of finding an approximate solution to $\sum_j w_j C_j$. The reduction is not approximation-preserving, but it comes pretty close. We show that any ρ-appr-oximation algorithm for minimizing total weighted completion time is a $(\rho + 1)$-approximation algorithm for the problem of minimizing $\alpha|\beta|\sum_j w_j(E_j + C_j)$. We leave on purpose undefined the specific setting, i.e., the number and type of machines, existence or not of release dates, precedence constraints and so on. Our approach is general enough to handle a number of variations. Using the 3-fleld scheduling notation we examine problems belonging to the family $\alpha|\beta|\sum_j w_j(E_j + C_j)$.

For any schedule σ let $C_j(\sigma)$, $E_j(\sigma)$ denote the completion time and earliness of job j in schedule σ. $\mathcal{E}(\sigma)$ denotes the set of early jobs, i.e., the jobs j for which $\mathcal{E}_j(\sigma) > 0$. Let $\mathcal{T}(\sigma)$ denote the set of tardy jobs, i.e., $\mathcal{T}(\sigma) := N \setminus \mathcal{E}(\sigma)$. By the definition of earliness, $C_j(\sigma) > d_j$ for any $j \in \mathcal{T}(\sigma)$, and the following two relations hold for any σ

$$\sum_{j\in N} w_j\left(E_j(\sigma) + C_j(\sigma)\right) = \sum_{j\in\mathcal{T}} w_j C_j(\sigma) + \sum_{j\in\mathcal{E}} w_j d_j \tag{1}$$

$$\sum_{j\in N} w_j C_j(\sigma) \le \sum_{j\in\mathcal{T}} w_j C_j(\sigma) + \sum_{j\in\mathcal{E}} w_j d_j \tag{2}$$

Consider now a particular schedule (σ^ρ) that achieves a ρ-approximation for the objective function $\sum_{j\in N} w_j C_j$ and let σ_C^* and $\sigma_\mathcal{E}^*$ be optimal schedules for the weighted completion time and weighted earliness objectives, respectively.

Then by (2)

$$\sum_{j\in N} w_j C_j(\sigma^\rho) \le \rho \sum_{j\in N} w_j C_j(\sigma_C^*) \le \rho \sum_{j\in N} w_j C_j(\sigma_\mathcal{E}^*) \le \rho OPT \tag{3}$$

where by *OPT* we denote the optimum value for the objective function $\sum_{j\in N} w_j(E_j + C_j)$.

By (1) the objective value achieved by σ^ρ is

$$\sum_{j\in N} w_j\left(E_j(\sigma^\rho) + C_j(\sigma^\rho)\right) = \sum_{j\in\mathcal{T}(\sigma^\rho)} w_j C_j(\sigma^\rho) + \sum_{j\in\mathcal{E}(\sigma^\rho)} w_j d_j$$

By (3) we have that

$$\sum_{j\in\mathcal{T}(\sigma^\rho)} w_j C_j(\sigma^\rho) \le \rho OPT - \sum_{j\in\mathcal{E}(\sigma^\rho)} w_j C_j(\sigma^\rho)$$

Therefore

$$\sum_{j\in N} w_j\left(E_j(\sigma) + C_j(\sigma)\right) \le \rho OPT - \sum_{j\in\mathcal{E}(\sigma^\rho)} w_j C_j(\sigma^\rho) + \sum_{j\in\mathcal{E}(\sigma^\rho)} w_j d_j \le (\rho + 1) OPT$$

Thus we have proved the following meta theorem.

Table 1:

Problem	ratio for $\sum_j w_j(E_j+C_j)$	reference for $\sum_j w_jC_j$
$P\|r_j,pmtn\|\sum_j w_j(E_j+C_j)$	$2+\varepsilon$	[1]
$P\|r_j\|\sum_j w_j(E_j+C_j)$		
$Rm\|r_j,pmtn\|\sum_j w_j(E_j+C_j)$		
$Rm\|r_j\|\sum_j w_j(E_j+C_j)$		
$Q\|pmtn\|\sum_j w_j(E_j+C_j)$	2	[17]
$Q\|r_j\|\sum_j w_j(E_j+C_j)$	$2+\varepsilon$	[9]
$R\|\ \|\sum_j(E_j+C_j)$	2	[21, 7]
$R\|\ \|\sum_j w_j(E_j+C_j)$	$2.5+\varepsilon$	[35]
$R\|r_j\|\sum_j w_j(E_j+C_j)$	$3+\varepsilon$	

Approximation ratios ρ+1 obtained by Theorems 2.1 and 2.2 for various instantiations of $\alpha|\beta|\sum_j w_j(E_j+C_j)$. The references give the sources for the corresp-onding ρ-approximation ratio achieved for $\alpha|\beta|\sum_j w_jC_j$.

Theorem 2.1 *Consider a member* $\alpha_o|\beta_o|\sum_j w_jC_j$ *of the family of nonpreemptive scheduling prob-lems* $\alpha|\beta|\sum_j w_jC_j$ *for which there is a* ρ*-approximation algorithm. Then the same algorithm achieves a* $(\rho+1)$*-approximation for the problem* $\alpha_o|\beta_o|\sum_j w_j(E_j+C_j)$.

The following corollary is a particularly nice consequence of the theorem.

Corollary 2.1 *The Weighted Shortest Processing Time (WSPT) algorithm that orders the jobs in order of nonincreasing ratio* w_j/p_j *is a* 2*-approximation algorithm for* $1|\ |\sum_j w_j(E_j+C_j)$.

Proof. WSPT is well-known to be optimal for $1|\ |\sum_j w_jC_j$.

In general, Theorem 2.1 implies that we have a polynomial-time 2-approximation algorithm for *every* case of $\alpha|\beta|\sum_j w_j(E_j+C_j)$ whenever the corresponding case of $\alpha|\beta|\sum_j w_jC_j$ is optimally solvable in polynomial time. This implies, for example, the existence of a 2-approximation for $1|prec=series\ parallal|\sum_j w_j(E_j+C_j)$ [24, 31]. We summarize the most important conseq-uences of Theorem 2.1 and the upcoming Theorem 2.2 in Table 1. We omit from the table bounds that are improved in Section 3 with an LP-based approach.

The above reduction of total weighted earliness scheduling problems to total weighted completion time problems can be extended in three ways. First, the rephrasing of Theorem 2.1 with "*c*-competitive online algorithm" in place of "ρ-approximation algorithm" clearly holds as well. The second extension applies to preemptive scheduling.

Theorem 2.2 *Consider a member* $\alpha_o|\beta_o,pmtn|\sum_j w_jC_j$ *of the family of scheduling problems* $\alpha|\beta,pmtn|\sum_j w_jC_j$ *for which there is a* ρ*-approximation algorithm. Then the same algorithm achieves a* $(\rho+1)$*-approximation for the problems* $\alpha_o|\beta_o,pmtn|\sum_j w_j(E_j+C_j)$.

Proof. To prove the theorem we reason as follows. Equations (1) and (2) hold in the preemptive setting as well. Let *OPT* stand now for the optimum value of $\sum_j w_j(E_j+C_j)$ with preemption. Equation (3) continues to hold in the new setting. As a consequence we obtain that the very simple Shortest Weighted Remaining Processing Time (SWRPT) algorithm, which solves optimally $1|r_j,pmtn|\sum_j w_jC_j$, achieves a 2-approxim-ation for $1|r_j,pmtn|\sum_j w_j(E_j+C_j)$. A third extension of Theorem 2.1 results as follows. One could have a stochastic input where the vector **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 [29] which yields a feasible schedule for each realization **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

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

See e.g., [22, 36] 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 2.1 holds as well. A number of constant-factor approximation results of this type are shown in [30, 36] 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 $P|r_j, prec|E[\sum_j w_j(E_j + C_j)]$.

3 LP-based algorithms for the modified objective

In this section we show that it is possible to improve upon the guarantee of Theorem 2.1 and bring down the approximation ratio by 1 in cases where LP-based algorithms with certain characteristics are available for approximating the total weighted completion time. As in Section 2 we leave on purpose undefined the specific setting. Our approach is general enough to handle a number of variations on machine environment and job characteristics.

The *tardiness* of a job j in a schedule σ is defined as $T_j(\sigma) := max\{C_j(\sigma) - d_j, 0\}$. We will use mathematical programming formulations with variables E_j, T_j, C_j to denote respectively the earliness, tardiness, and completion time of job j, $j = 1, \ldots, n$. We propose a family of mathema-tical programs which is parameterized based on a set of constraints $\mathcal{C}(C)$. In principle the constraints in $\mathcal{C}(C)$ could be general convex. For our applications they will always be linear inequalities or equalities. A set of linear constraints $\mathcal{C}(C)$ is *a valid set of completion time constraints* if the completion times C_j $(j = 1,2, \ldots, n)$ must satisfy the constraints $\mathcal{C}(C)$ for *every* feasible schedule. As a concrete example for $\mathcal{C}(C)$, consider the problem $1|prec|\sum_j w_j(E_j + C_j)$, i.e., scheduling precedence-constrained jobs on a single machine. A valid set of completion time constraints, which was introduced by Queyranne [33], is the following:

$C_k > C_j + p_k$ for each pair j, k $s.t.$ $j \prec k$ (4)

$\sum_{j\in S} p_j C_j \geq \frac{1}{2}(p^2(S) + p(S)^2) \quad \forall\, S \subseteq N$ (5)

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 (5) 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 by Potts [32] and Chudak and Hochbaum [13] or Margot, Queyranne and Wang [28] and the time-indexed formulation of Dyer and Wolsey [16]. 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 $\alpha|\beta|\sum_j w_j(E_j + C_j)$. We emphasize that the E_j or T_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 earliness and tardiness variables do not appear in $\mathcal{C}(C)$. We propose the following family of linear programs, denoted $FP(\mathcal{C})$:

$minmize \sum_{j=1}^{n} w_j(E_j + C_j)$ (6)

$E_j = d_j - C_j + T_j \quad j = 1, \ldots, n$ (7)

$\mathcal{C}(C)$ (8)

$E_j, T_j, C_j \geq 0 \quad j = 1, \ldots, n$ (9)

Since every job is either tardy or early in any feasible schedule, i.e., at most one of the two variables E_j and T_j can be positive, it is easy to see that the E_j and T_j values must satisfy (7) for any feasible schedule. Of course equations (7) do allow solutions in which both E_j and T_j are positive, thus $FP(\mathcal{C})$: is normally not an exact formulation for the problem $\alpha|\beta|\sum_j w_j(E_j + C_j)$, it is only a linear programming relaxation. As another example consider the case where the jobs also have release dates, i.e., $1|r_j, prec|\sum_j w_j(E_j + C_j)$. Hall, Schulz, Shmoys and Wein [20] studied the valid set of completion time constraints made up of (4), (5) and the following inequalities $C_j \geq r_j + p_j \quad j = 1, \ldots, n$ Substituting these constraints for $\mathcal{C}(C)$ in the above formulation $FP(\mathcal{C})$ yields a linear programming relaxation for $1|r_j, prec|\sum_j w_j(E_j + C_j)$. Our results are based on the following algorithm schema for the generic problem $\alpha|\beta|\sum_j w_j(E_j + C_j)$. 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 $C_j(\sigma) < \rho\overline{C}_j$, $j = 1, \ldots, n$, for some $\rho \geq 1$.

ALGORITHM SCHEMA($\mathcal{C}$)
1. Compute an optimal solution to $FP(\mathcal{C})$. Let $\overline{E}_j, \overline{T}_j, \overline{C}_j$ 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 $C_j < \rho\overline{C}_j$, $j = 1, \ldots, n$, for some $\rho \geq 1$.
3. Output the schedule σ.

Consider the problem $1|prec|\sum_j w_j(E_j + C_j)$. As an example for the algorithm $\mathcal{A}(\alpha, \beta)$ for this problem, we mention the *Schedule-by-*$\overline{C}_j$ heuristic from [20], which constructs the schedule σ by sequencing the jobs in nondecreasing order of the $\overline{C}_j$. It is well-known that this yields a schedule σ with $C_j(\sigma) < 2\overline{C}_j$, $j = 1, \ldots, n$.

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

Lemma 3.1 *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* $\alpha|\beta|\sum_j w_j(E_j + C_j)$.

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

$$E_j(\sigma) = d_j - C_j(\sigma) + T_j(\sigma) \leq d_j - \rho\overline{C}_j + T_j(\sigma). \qquad (10)$$

If job j is not early, $E_j(\sigma) = 0$ and the contribution of job j to the objective is $\sum_j w_j C_j$. If job j is early, the contribution of j to the objective is $w_j\left(E_j(\sigma) + C_j(\sigma)\right)$ which by (10) is at most

$$w_j\left(E_j(\sigma) + C_j(\sigma)\right) \leq w_j(\rho\overline{C}_j + d_j) = \rho w_j\overline{C}_j$$

But from (7)

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

Hence

$$w_j\left(E_j(\sigma) + C_j(\sigma)\right) = w_j d_j \leq \rho w_j\overline{C}_j \leq \rho w_j\left(\overline{E}_j(\sigma) + C_j(\sigma)\right)$$

and the lemma is shown.

Remark 3.1 *The analysis of Lemma* 3.1 *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.* It suffices to identify problems of the form $\alpha|\beta|\sum_j w_j(E_j+C_j)$ for which (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 described exists. One can mine the seminal paper of Hall, Schulz, Shmoys and Wein [20] and subsequent literature for a wealth of related results. The main problems for which our requirements are satisfied are shown in Table 2.

Using Lemma 3.1 we obtain the following meta theorem.

Theorem 3.1 *There are constant-factor ρ-approximation algorithms for the family of scheduling problems $\alpha|\beta|\sum_j w_j(E_j+C_j)$ where α stands for* 1 *or P and β for all possible combinations of r_j, prec, pmtn. The values of ρ for each case are given in Table* 2.

Table 2:

Problem	ratio for $\sum_j w_j(E_j+C_j)$	reference for $\sum_j w_j C_j$
$1\|prec\|\sum_j w_j(E_j+C_j)$	ρ = 2	[20]
$1\|r_j, prec, pmtn\|\sum_j w_j(E_j+C_j)$	ρ = 2	
$1\|r_j, prec\|\sum_j w_j(E_j+C_j)$	ρ= 3	
$P\| \|\sum_j w_j(E_j+C_j)$	ρ = 2 – 1/*m*	
$P\|r_j, prec, pmtn\|\sum_j w_j(E_j+C_j)$	ρ = 3	
$P\|r_j, prec\|\sum_j w_j(E_j+C_j)$	ρ = 4	[31]
$Q\|r_j, prec, pmtn\|\sum_j w_j(E_j+C_j)$	ρ = *O*(log *m*)	[14]
$Q\|r_j, prec\|\sum_j w_j(E_j+C_j)$		

Approximation ratios achieved by Theorems 3.1 and 3.2 for various instantiations of $\alpha|\beta|\sum_j w_j(E_j+C_j)$. The references give the sources for the corresponding $\mathcal{A}(\alpha,\beta)$ subroutine required by the Algorithm Schema.The results of Chudak and Shmoys [14] for the problems $Q|r_j, prec|\sum_j w_j C_j$ and $Q|r_j, prec, pmtn|\sum_j w_j C_j$ provide the valid set of completion time constraints together with an *O*(*log m*)-approximation algorithm, *m* being the number of machines, with a job-by-job guarantee. We therefore have

Theorem 3.2 *Let m be the number of machines. There are O*(log *m*)*-approximation algorithms for $Q|r_j, prec|\sum_j w_j(E_j+C_j)$ and $Q|r_j, prec, pmtn|\sum_j w_j(E_j+C_j)$.*

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 $\alpha|\beta|\sum_j w_j(E_j+C_j)$ with $E_j=0$ for $j=1,\dots,n$, 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 $\alpha|\beta|\sum_j w_j(E_j+C_j)$ problem.

4 An FPTAS for the common due date case

In this section we will present an FPTAS for the problem $1|d_j=D|\sum_j w_j(E_j+C_j)$, 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 [38]. Lawler and Moore [25] 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 [25] and [23], we are going to scale and round down the processing times by a constant K, which is to be determined later. Unlike [25] and [23], 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 $j = 1,\ldots,n$. 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 $\sum_j w_j E_j$ and denote by $E(\sigma^*)$ this optimum value. Let $\overline{E}_{\sigma_A(j)}$ be the earliness 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 $\overline{E}_{\sigma_A(j)}$ be the earliness of the same job in σ_A with the original data. In addition, define $E_{\sigma_A} := \sum_{j=1}^n w_{\sigma_A(j)} E_{\sigma_A(j)}$. Then we clearly have $\overline{E}_{\sigma_A(j)} \le E_{\sigma_A(j)}/K$ for $j = 1,\ldots,n$. Furthermore, $\overline{E}_{\sigma_A} := \sum_{j=1}^n w_{\sigma_A(j)} \overline{E}_{\sigma_A(j)} \le E(\sigma^*)/K$ since σ_A is optimal for the scaled down data. Let E'_{σ_A} denote the total weighted earliness of the sequence σ_A when we use processing times $p' := K\overline{p}_j$ for each job j and the original due dates d_j. Note that $p' := K\overline{p}_j \le p_j \le K\left(\overline{p}_j + 1\right)$. We can write

$$K\overline{E}_{\sigma_A} \le E(\sigma^*) \le E_{\sigma_A} \le \sum_{j=1}^n w_{\sigma_A(j)}\, max\left\{d_{\sigma_A(j)} - K\sum_{i=1}^j \left(\overline{p}_{\sigma_A(i)} + 1\right), 0\right\}$$

$$\le E'_{\sigma_A} + Kn\sum_{j=1}^n w_j \qquad (11)$$

Furthermore,

$$\begin{aligned} K\overline{E}_{\sigma_A} &= K\sum_{j=1}^n w_{\sigma_A(j)}\, max\left\{\overline{d}_{\sigma_A(j)} - K\sum_{i=1}^j \overline{p}_{\sigma_A(i)}, 0\right\} \\ &\ge K\sum_{j=1}^n w_{\sigma_A(j)}\, max\left\{\left(\frac{d_{\sigma_A(j)}}{K} + 1\right) - K\sum_{i=1}^j \overline{p}_{\sigma_A(i)}, 0\right\} \\ &\ge \sum_{j=1}^n w_{\sigma_A(j)}\, max\left\{d_{\sigma_A(j)} - K\sum_{i=1}^j \overline{p}_{\sigma_A(i)}, 0\right\} - K\sum_{j=1}^n w_{\sigma_A(j)} = E'_{\sigma_A} - K\sum_{j=1}^n w_j \qquad (12) \end{aligned}$$

Combining (11) and (12), we obtain

$$E'_{\sigma_A} - K\sum_{j=1}^n w_j \le E(\sigma^*) \le E_{\sigma_A} \le E'_{\sigma_A} + Kn\sum_{j=1}^n w_j$$

which implies

$$E_{\sigma_A} - E(\sigma^*) \le K(n+1)\sum_{j=1}^n w_j \qquad (13)$$

Choose $K = \varepsilon D/(n+1)$. By (13) the error is at most $\varepsilon \sum_j w_j D = \varepsilon \sum_j w_j d_j \le \varepsilon OPT$. The running time of the algorithm that computes σ_A on the scaled input is $O\left(n^2(D/K)\right) = O(n^3/\varepsilon)$. We have proved the following theorem.

Theorem 4.1 *There is an FPTAS for the problem* $1\left|d_j = D\right|\sum_j w_j\left(E_j + C_j\right)$, *i.e., minimizing* $\sum_j w_j\left(E_j + C_j\right)$ *on a single machine when all the jobs have a common due date.*

References :

[1] F. Afrati, E. Bampis, C. Chekuri, D. Karger, C. Kenyon, S. Khanna, I. Milis, M. Queyranne, M. Skutella, C. Stein, and 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, 32–43, 1999.
[2] R., Ahmadi and U. Bagchi. Single machine scheduling to minimize earliness subject to deadlines. Working Paper 86/86-4-17, Department of Management, University of Texas, Austin, 1986.
[3] R. Ahmadi and U. Bagchi. Just-in-time scheduling in single machine systems, Working Paper 85/86-4-21, Department of Management, University of Texas, Austin, 1986.
[4] M. G. Ahmed. Earliness Penalties on a Single Machine. Msc. Thesis, Univ. of Al- Mustans-iriyah, College of Science, Dept. of Mathematics, 2005.
[5] M. Asano and H. Ohta. Single machine scheduling using dominance relation to minimize earliness subject to ready and due times. International Journal of Production Economics, 44: 35-43,1996.
[6] P. Brucker. Scheduling algorithms. Springer Berlin Heidelberg New York, 2007.
[7] J.L. Bruno, E.G. Coffman Jr, and R. Sethi. Scheduling independent tasks to reduce mean finishing time. Communications of the ACM, 17:382–387, 1974.
[8] S. Chand and Schneebergerh. Single machine scheduling to minimize weighted earliness subject to no tardy jobs. European Journal of Operational Research, 34: 221-230, 1988.
[9] C. Chekuri and S. Khanna. A PTAS for minimizing weighted completion time on uniformly related machines. In Proceedings of the 28th ICALP, 848–861, 2001.
[10] C. Chekuri and S. Khanna. Approximation algorithms for minimizing the weighted sum of completion times. In J. Y-T. Leung, editor, Handbook of Scheduling: Algorithms, Models, and Performance Analysis. Chapman & Hall/CRC, 2004.
[11] T. C. E. Cheng, C. T. Ng, J. J. Yuan, and Z. H. Liu. Single machine scheduling to minimize total weighted tardiness. European Journal of Operational Research, 165:423–443, 2005.
[12] E. Chhajed. A fixed interval deadline scheduling problem with earliness and deadline costs. European Journal of Operational Research, 84: 385-401,1995.
[13] F. A. Chudak and D. S. Hochbaum. A half-integral linear programming relaxation for scheduling precedence-constrained jobs on a single machine. Operations Research Letters, 25:199–204, 1999.
[14] F. A. Chudak and D. B. Shmoys. Approximation algorithms for precedence-constrained scheduling problems on parallel machines that run at different speeds. Journal of Algorithms, 30:323–343, 1999.
[15] J. Du and Leung. Minimizing total tardiness on one processor is NP-hard, Mathematics of Operations Research. 15:483-495, 1990.
[16] M.E. Dyer and L.A. Wolsey. Formulating the single machine sequencing problem with release dates as a mixed integer program. Discrete Applied Mathematics, 26:255–270, 1990.
[17] T. Gonzalez. Optimal mean finish time preemptive schedules. Technical Report Tech. Rep. 220, Comp. Sci. Dept., Pennsylvania State University, 1977.
[18] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics, 5:287–326, 1979.
[19] L. A. Hall. Approximation algorithms for scheduling. In D. S. Hochbaum, editor, Approximation Algorithms for NP-hard problems, 1–45. PWS, 1997.
[20] 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.
[21] W. Horn. Minimizing average flow time with parallel machines. Operations Research, 21:846–847, 1973.

[22] H. Kellerer, T. Tautenhahn, and 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, 418–426, 1996.
[23] S. G. Kolliopoulos and G. Steiner. Approximation algorithms for minimizing the total weighted tardiness on a single machine. Submitted for journal publication. Preliminary version in Proceedings of STACS 04.
[24] E. L. Lawler. Sequencing jobs to minimize total weighted completion time subject to precedence constraints. Annals of Discrete Mathematics, 2:75–90, 1978.
[25] E. L. Lawler. A fully polynomial approximation scheme for the total tardiness problem. Operations Research Letters, 1:207–208, 1982.
[26] E. L. Lawler and J. M. Moore. A functional equation and its application to resource allocation and sequencing problems. Management Science, 16:77–84, 1969.
[27] J.K. Lenstra, A.H.G. Rinnooy Kan, and P. Brucker. Complexity of machine scheduling problems. Annals of Discrete Mathematics, 1:343–362, 1977.
[28] F. Margot, M. Queyranne, and Y. Wang. Decompositions, network flows and a precedence constrained single-machine scheduling problem. Operations Research, 51(2000-29):981–992, 2003.
[29] R. H. Möhring, F. J. Radermacher, and G. Weiss. Stochastic scheduling problems I: general strategies. ZOR - Zeitschrift für Operations Research, 28:193–260, 1984.
[30] R. H. Möhring, A. S. Schulz, and M. Uetz. Approximation in stochastic scheduling: the power of LP-based priorities. Journal of the ACM, 46:924–942, 1999.
[31] A. Munier, M. Queyranne, and 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, 367–382, 1998.
[32] C. N. Potts. An algorithm for the single machine sequencing problem with precedence constraints. Mathematical Programming Study, 13:78–87, 1980.
[33] M Queyranne. Structure of a simple scheduling polyhedron. Mathematical Programming, 58:263–285, 1993.
[34] Y.A. Rios-Solis, F. Sourd. Exponential neighborhood search for a parallel machine scheduling problem. Computers and Operations Research, 35(5):1697 – 1712, 2008.
[35] A. Schulz and M. Skutella. Scheduling unrelated machines by randomized rounding. SIAM Journal on Discrete Mathematics, 15:450–469, 2002.
[36] M. Skutella and M. Uetz. Stochastic machine scheduling with precedence constraints. SIAM Journal on Computing, 34:788–802, 2005.
[37] W.E. Smith. Various optimizers for single – stage production. Naval. Res. Logist Q. 3:56-66, 1956.
[38] J. Yuan. The NP-hardness of the single machine common due date weighted tardiness problem. Systems Science and Mathematical Sciences, 5:328–333, 1992.