J Sched (2008) 11: 217–228
DOI 10.1007/s10951-007-0015-y

Optimal due date assignment in multi-machine scheduling environments

Dvir Shabtay · George Steiner

Received: 29 June 2006 / Accepted: 3 May 2007 / Published online: 13 July 2007

Abstract We study two due date assignment problems in various multi-machine scheduling environments. We assume that each job can be assigned an arbitrary non-negative due date, but longer due dates have higher cost. The first problem is to minimize a cost function, which includes earliness, tardiness and due date assignment costs. In the second problem, we minimize an objective function which includes the number of tardy jobs and due date assignment costs. We settle the complexity of many of these problems by either showing that they are $\mathcal{NP}$-hard or by providing a polynomial time solution for them. We also include approximation and non-approximability results for several parallel-machine problems.

Keywords Multi-machine scheduling · Due date assignment · Complexity · Approximation

1 Introduction

Meeting due dates has always been one of the most important objectives in scheduling and supply chain management. Customers demand that suppliers meet contracted delivery dates or face large penalties. For example, Slotnick and Sobel (2005) cite contracts from the aerospace industry, which may impose tardiness penalties as high as one million dollars per day on subcontractors for aircraft components. Traditional scheduling models considered due dates as given by exogenous decisions (see Baker and Scudder 1990 for a survey). In an integrated system, however, they are determined by taking into account the system's capacity to meet the quoted delivery dates. In order to avoid tardiness penalties, companies are under increasing pressure to quote attainable delivery dates. At the same time, promising delivery dates too far into the future may not be acceptable to the customer or may force a company to offer price discounts in order to retain the business. Thus, there is an important trade-off between assigning relatively short due dates to customer orders and avoiding tardiness penalties. This is why an increasingly large number of recent studies viewed due date assignment as part of the scheduling process, and showed how the ability to control due dates can be a major factor in improving system performance.

Early research in the area of due date assignment in scheduling was due to Seidmann et al. (1981) and Panwalkar et al. (1982). Panwalkar et al. (1982) studied the *constrained version*, where the scheduler must decide on a *common* due date for all jobs (this method is usually abbreviated as the CON due date assignment method), while Seidmann et al. (1981) dealt with the *unrestricted case*, where each job can have a *different* due date (we will refer to this due date assignment method as DIF). These two papers started extensive research in the area of due date assignment, with most papers focusing on the common due date assignment problem (e.g., Bagchi et al. 1986a, 1986b; De et al. 1991; Kahlbacher and Cheng 1993; Cheng and Kovalyov 1996;

This research was partially supported by the Paul Ivanier Center for Robotics and Production Management, Ben-Gurion University of the Negev. Partial support by the Natural Sciences and Engineering Research Council of Canada under Grant No. 041798 is also gratefully acknowledged.

D. Shabtay (✉) · G. Steiner
Management Science and Information Systems Area, Michael G. DeGroote School of Business, McMaster University, Hamilton, ON, Canada
e-mail: dvirs@bgu.ac.il

G. Steiner
e-mail: steiner@mcmaster.ca

Mosheiov 2001; Birman and Mosheiov 2004). A recent survey on common due date assignment problems was given by Gordon et al. (2002).

In this paper we study scheduling problems in a multi-machine environment, where the due dates are assignable according to the DIF due date assignment method. As far as we know, this method was studied only for the single machine scheduling problem by Seidmann et al. (1981) and Shabtay and Steiner (2005). We study the problems of minimizing two different objective functions. The first objective is to minimize the sum of due date assignment, earliness and tardiness penalties as given by the following objective function

$$\sum_{i=1}^{n}\bigl(\alpha \max(0, d_i - A) + \beta E_i + \gamma T_i\bigr), \tag{1}$$

where C_i is the *completion time* of job i, $E_i = \max(0, d_i - C_i)$ is the *earliness* of job i, $T_i = \max(0, C_i - d_i)$ is the *tardiness* of job i, $A \geq 0$ represents the *lead time* that customers consider to be acceptable, and α, β and γ are non-negative parameters representing the per unit lead-time, earliness and tardiness penalties, respectively. There is no lead-time cost if the due date is set to be less than or equal to A.

Our second objective is to minimize the sum of the due date assignment costs and the number of tardy jobs given by the objective function

$$\sum_{i=1}^{n}\bigl(\alpha \max(0, d_i - A) + \beta U_i\bigr), \tag{2}$$

where U_i is the tardiness indicator variable for job i, i.e., $U_i = 1$ if $C_i > d_i$ and $U_i = 0$ if $C_i \leq d_i$, and β is the cost of a tardy job i.

In order to specify each problem, we use the classical 3-field notation introduced in Graham et al. (1979). Since α, β and γ are usually reserved for the cost coefficients in the due date assignment literature, we use $X|Y|Z$ to refer to the 3 fields. The X field describes the machine environment: $X \in \{Pm, Qm, Rm, Fm, Jm, Om\}$ for identical parallel, uniform or unrelated machines, flow shops, job shops or open shops, respectively.

The Y field exhibits the job-processing characteristics and constraints and may contain no entry, a single entry, or multiple entries. For example, if DIF or CON appear in the Y field, this means that the due dates are assignable according to the DIF or CON due date assignment method, respectively. We denote the *processing time of job i* by p_i. In the case of uniform and unrelated machines the *actual processing time of job i* on machine j is p_i/s_j and p_i/s_{ij}, respectively, where s_j and s_{ij} denote the speed of machine j in the respective cases. For the complexity analysis, we assume, without loss of generality, that all job-related data, i.e., processing times, lead times and due dates are non-negative integers.

The Z field contains the objective function for the scheduling problem, and in this paper, it will usually refer to one of the two objective functions defined in (1) or (2).

Seidmann et al. (1981) presented an $O(n \log n)$ algorithm to solve the $1|DIF|\sum_{i=1}^{n}(\alpha \max(0, d_i - A) + \beta E_i + \gamma T_i)$ problem. We study the complexity of the $X|DIF|\sum_{i=1}^{n}(\alpha \max(0, d_i - A) + \beta E_i + \gamma T_i)$ problems with $X \in \{Pm, Qm, Rm, Fm, Jm, Om\}$. All these problems are proven to be $\mathcal{NP}$-hard, but there are several important special cases, which can be solved in polynomial time. In particular, we give polynomial time solutions for the case of *no acceptable lead times*, i.e., when $A = 0$, on parallel machines. For $\rho \geq 1$, an algorithm H is a ρ-approximation algorithm for a scheduling problem if, for any instance of the problem, it is guaranteed to find a schedule whose cost is at most ρ times the minimum cost. A family of algorithms $\{H_\varepsilon\}$ for a problem is called a *fully polynomial time approximation scheme (FPTAS)* if, for every $\varepsilon > 0$, H_ε is a $(1+\varepsilon)$-approximation algorithm whose running time is polynomial in the input size and $1/\varepsilon$ (Garey and Johnson 1979). We also study the approximability of the $X|DIF|\sum_{i=1}^{n}(\alpha \max(0, d_i - A) + \beta E_i + \gamma T_i)$ problems for $A > 0$ on parallel machines. We show that these problems cannot have a polynomial time ρ-approximation algorithm with $\rho < \infty$ unless $\mathcal{P} = \mathcal{NP}$. We also prove that if we modify the objective by adding an appropriate $b > 0$ to it, then there is a polynomial time 2-approximation algorithm for the $Rm|DIF|\sum_{i=1}^{n}(\alpha \max(0, d_i - A) + \beta E_i + \gamma T_i) + b$ problem. A summary of the known and the new results in this paper is given in Table 1.

Shabtay and Steiner (2005) presented an $O(n \log n)$ optimization algorithm to solve the $1|DIF|\sum_{i=1}^{n}(\alpha \max(0, d_i - A) + \beta U_i)$ problem. In this paper we show that for all multi-machine scheduling environments this problem becomes $\mathcal{NP}$-hard, but the case of no acceptable lead times ($A = 0$) becomes polynomially solvable again on parallel machines. We also present approximation and non-approximability results for these problems when $A > 0$. A summary of the results is again contained in Table 1.

The paper is organized as follows. In Sect. 2 we prove that the $X|DIF|\sum_{i=1}^{n}(\alpha \max(0, d_i - A) + \beta E_i + \gamma T_i)$ problem is $\mathcal{NP}$-hard for all multi-machine scheduling environments. While it is strongly $\mathcal{NP}$-hard for $X = Fm$, $X = Jm$ and $X = Om$, we provide a pseudo-polynomial algorithm to solve the problem on identical and uniform parallel machines with a fixed number of machines. We also prove that with $A = 0$ the problem has a polynomial time solution on parallel machines. This is followed by our non-approximability results for the $A > 0$ case and approximation algorithms for a modified objective. In Sect. 3 we present similar results for the $X|DIF|\sum_{i=1}^{n}(\alpha \max(0, d_i -$

Table 1 Summary of results

Problem	Complexity	Reference
$1\|DIF\|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$	$O(n\log n)$	(Seidmann et al. 1981)
$Pm\|DIF\|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$	$\mathcal{NP}$-hard in the ordinary sense	Theorem 1 and Sect. 2.2
$Qm\|DIF\|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$	$\mathcal{NP}$-hard in the ordinary sense	Theorem 1 and Sect. 2.2
$Rm\|DIF\|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$	$\mathcal{NP}$-hard	Theorem 1
$Pm\|DIF\|\sum_{i=1}^n(\alpha d_i+\beta E_i+\gamma T_i)$	$O(n\log n)$	Corollary 3
$Pm\|CON\|\sum_{i=1}^n(\alpha d_i+\beta E_i+\gamma T_i)$	$\mathcal{NP}$-hard	(Cheng and Chen 1994; De et al. 1994)
$Rm\|DIF\|\sum_{i=1}^n(\alpha d_i+\beta E_i+\gamma T_i)$	$O(n^3)$	Corollary 3
$Fm\|DIF\|\sum_{i=1}^n(\alpha d_i+\beta E_i+\gamma T_i)$	Strongly $\mathcal{NP}$-hard	Theorem 1
$Om\|DIF\|\sum_{i=1}^n(\alpha d_i+\beta E_i+\gamma T_i)$	Strongly $\mathcal{NP}$-hard	Theorem 1
$Pm\|DIF\|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$	no ρ-approx. with $\rho<\infty$	Theorem 2
$Rm\|DIF\|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)+wA$	2-approx.	Corollary 2
$1\|DIF\|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)$	$O(n\log n)$	(Shabtay and Steiner 2005)
$Pm\|DIF\|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)$	$\mathcal{NP}$-hard in the ordinary sense	Theorem 5 and Sect. 3.2
$Qm\|DIF\|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)$	$\mathcal{NP}$-hard in the ordinary sense	Theorem 5 and Sect. 3.2
$Rm\|DIF\|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)$	$\mathcal{NP}$-hard	Theorem 5
$Pm\|DIF\|\sum_{i=1}^n(\alpha d_i+\beta U_i)$	$O(n\log n)$	Corollary 6
$Pm\|CON\|\sum_{i=1}^n(\alpha d_i+\beta U_i)$	$\mathcal{NP}$-hard	(Kahlbacher and Cheng 1993)
$Rm\|DIF\|\sum_{i=1}^n(\alpha d_i+\beta U_i)$	$O(n^{m+3})$	Theorem 10
$Fm\|DIF\|\sum_{i=1}^n(\alpha d_i+\beta U_i)$	Strongly $\mathcal{NP}$-hard	Theorem 5
$Om\|DIF\|\sum_{i=1}^n(\alpha d_i+\beta U_i)$	Strongly $\mathcal{NP}$-hard	Theorem 5
$Pm\|DIF\|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)$	no ρ-approx. with $\rho<\infty$	Theorem 6
$Rm\|DIF\|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i+\alpha A)$	2-approx.	Corollary 5

$A)+\beta U_i)$ problem following the same outline. A concluding summary is given in the last section.

2 The $X|DIF|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$ problem

In this section we analyze the $X|DIF|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$ problem for $X\in\{Pm,Qm,Rm,Fm,Jm,Om\}$. For given predefined due dates, the problem is $\mathcal{NP}$-hard even in the single-machine case with $\beta=0$, as it becomes the well-known $1||\sum T_i$ problem (see Du and Leung 1990).

2.1 Analysis and complexity

For any given schedule, which fixes the set of completion times $\mathbf{C}=(C_1,C_2,\ldots,C_n)$, each of the problems $X|DIF|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$ is reduced to a pure due date assignment problem, that is, the determination of the set of due dates, $\mathbf{d}=(d_1,d_2,\ldots,d_n)$, which minimizes the objective given by (1) with every other variable fixed. It is easy to see from (1) that this due date assignment problem has a separable objective function. Thus, we can determine the optimal due date for job i by determining d_i that minimizes the following objective for $i=1,\ldots,n$:

$$Z_i(d_i)=\alpha\max(0,d_i-A)+\beta\max(0,d_i-C_i)+\gamma\max(0,C_i-d_i). \quad (3)$$

This due date assignment problem was analyzed by Shabtay and Steiner (2005) in the single-machine case. However, *if* the completion times are fixed, the problem no longer depends on the machine environment. Thus, the same arguments can be used as in the single-machine case, and, therefore, we present the results of this analysis in the following lemma without a proof.

Lemma 1 *For any of the multi-machine scheduling problems* $X|DIF|\sum_{i=1}^{n}(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$ *with a* fixed *set of completion times* $\mathbf{C}=(C_1,C_2,\dots,C_n)$, *the optimal due date assignment policy* $\mathbf{d}^*=(d_1^*,\dots,d_n^*)$ *is as follows: If* $C_i\le A$ *then set* $d_i^*=C_i$; *otherwise, if* $\alpha<\gamma$ *then set* $d_i^*=C_i$, *and if* $\alpha\ge\gamma$ *then set* $d_i^*=A$.

By Lemma 1, under an optimal due date assignment strategy, we have $Z_i=0$ for any job i with $C_i\le A$, and $Z_i=w\times(C_i-A)$ for any job i with $C_i\ge A$, where $w=\min(\alpha,\gamma)$. Therefore, under an optimal due date assignment strategy, (1) becomes

$$Z(\mathbf{C},\mathbf{d}^*)=w\sum_{i=1}^{n}\max(0,C_i-A). \tag{4}$$

Theorem 1 *The* $X|DIF|\sum_{i=1}^{n}(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$ *problem is equivalent to a corresponding* $X|d_i=A|\sum_{i=1}^{n}T_i$ *problem with fixed common due date* A *for* $X\in\{Pm,Qm,Rm,Fm,Jm,Om\}$. *Furthermore, the* $X|DIF|\sum_{i=1}^{n}(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$ *problem is* $\mathcal{NP}$*-hard for* $X\in\{Pm,Qm,Rm\}$. *The problem is strongly* $\mathcal{NP}$*-hard for* $X\in\{Fm,Jm,Om\}$ *even when* $A=0$.

Proof It is easy to see that the objective function in (4) has the format of a sum-of-tardiness objective with given common due date A. Therefore, under an optimal due date assignment strategy the problem $X|DIF|\sum_{i=1}^{n}(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$ is equivalent to the corresponding $X|d_i=A|\sum_{i=1}^{n}T_i$ problem with implied non-assignable common due date $d_i=A$ for $i=1,\dots,n$. Therefore, in order to prove that $X|DIF|\sum_{i=1}^{n}(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$ is $\mathcal{NP}$-hard it is sufficient to show that the corresponding $X|d_i=A|\sum_{i=1}^{n}T_i$ problem is $\mathcal{NP}$-hard.

The $Pm|d_i=A|\sum_{i=1}^{n}T_i$ problem is known to be $\mathcal{NP}$-hard for $m=2$, since the problem of finding a schedule with value $\sum_{i=1}^{n}T_i=0$ for the instance where $A=1/2\times\sum_{i=1}^{n}p_i$ is equivalent to the $\mathcal{NP}$-hard problem MULTIPROCESSOR SCHEDULING (see Garey and Johnson 1979). Since $Pm|d_i=A|\sum_{i=1}^{n}T_i$ is a special case of the $Qm|d_i=A|\sum_{i=1}^{n}T_i$ and the $Rm|d_i=A|\sum_{i=1}^{n}T_i$ problems, it is straightforward to see that the last two are also $\mathcal{NP}$-hard. For $A=0$, the $X|d_i=0|\sum_{i=1}^{n}T_i$ problem is equivalent to a $X||\sum_{i=1}^{n}C_i$ problem. This problem is known to be strongly $\mathcal{NP}$-hard for $X=F2$ and $X=J2$ (see Garey et al. 1976) and also for $X=O2$ (see Achugbue and Chin 1982). □

2.2 Pseudo-polynomial algorithms for identical and uniform parallel machines

The following lemma gives a useful property of an optimal schedule for parallel-machine scheduling problems.

Lemma 2 *There exists an optimal schedule for* $Rm|DIF|\sum_{i=1}^{n}(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$, *in which the jobs are sequenced according to the shortest processing time* (*SPT*) *rule on each machine.*

Proof Any given job assignment to machines defines the actual job processing times. Thus, our scheduling problem $Rm|DIF|\sum_{i=1}^{n}(\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i)$, under an optimal due date assignment strategy and fixed job-to-machine assignment, reduces to m unrelated $1|d_i=A|\sum_{i=1}^{n}T_i$ problems, i.e., we have to solve m unrelated single-machine problems with a common non-assignable due date A to minimize the total tardiness. It is well-known that $1|d_i=A|\sum_{i=1}^{n}T_i$ is minimized by sequencing the jobs in a non-decreasing order of processing times, i.e., according to the SPT rule. □

Rothkopf (1966) and Lawler and Moore (1969) have suggested a general dynamic programming optimization algorithm for a fixed number of machines, which is applicable to special cases of $Rm||\sum_{i=1}^{n}f_i$, where f_i is a *regular* (non-decreasing) criterion for $i=1,\dots,n$ and it is possible to index the jobs in such a way that the jobs assigned to a given machine can be assumed to be processed in order of their indices. The algorithm can be described as follows.

Given an appropriate indexing $i=1,\dots,n$ of the jobs, define $F_i(t_1,\dots,t_m)$ as the minimum cost of a schedule for jobs $J_1,\dots,J_i$ subject to the constraint that the last job on M_j is completed at time t_j for $j=1,\dots,m$. Then for the $\sum_{i=1}^{n}f_i$ criteria, we have

$$F_i(t_1,\dots,t_m)=\min_{j=1,\dots,m}\{F_{i-1}(t_1,\dots,t_j-p_{ij},\dots,t_m)+f_i(t_j)\}; \tag{5}$$

the initial conditions are

$$F_0(t_1,\dots,t_m)=\left\{\begin{matrix}0, & \text{if } t_j=0 \text{ for } j=1,\dots,m\\ \infty, & \text{otherwise}\end{matrix}\right\}; \tag{6}$$

and the optimal solution value is given by

$$F_n^*=\min\big(F_n(t_1,\dots,t_m)\mid 0\le t_j\le C\big), \tag{7}$$

where C is an upper bound on the completion time of any job in an optimal schedule. In general, these equations can be solved in $O(mnC^m)$ time, but if the machines are uniform, only $m-1$ of the $t_1,\dots,t_m$ values are independent.

This means that for uniform machines, the time complexity reduces to $O(mnC^{m-1})$.

We will show that a variant of the above optimization algorithm solves the $Qm|d_i = A|\sum_{i=1}^n T_i$ and the $Pm|d_i = A|\sum_{i=1}^n T_i$ problems that our $Qm|DIF|\sum_{i=1}^n(\alpha\max(0, d_i - A) + \beta E_i + \gamma T_i)$ and $Pm|DIF|\sum_{i=1}^n(\alpha\max(0, d_i - A) + \beta E_i + \gamma T_i)$ problems are equivalent to, respectively. First we have to note that $\sum_{i=1}^n T_i$ is a *regular* criterion. In addition, we know that there exists an optimal schedule where the jobs on each machine are in SPT order according to Lemma 2. It is easy to see that for identical or uniform parallel machines, the SPT order of the jobs is the *same* no matter which machine we consider. (Note, however, that this is not necessarily the case for the $Rm|d_i = A|\sum_{i=1}^n T_i$ problem.) Therefore, the SPT order can serve as the common indexing of the jobs required by the above dynamic programming algorithm. We can apply (5–7) with

$$f_i(t_j) = \left\{ \begin{array}{ll} 0, & \text{if } t_j \le A \\ t_j - A, & \text{if } t_j > A \end{array} \right\},$$

$$\text{for } i = 1, \dots, n \text{ and } j = 1, \dots, m. \tag{8}$$

It is clear that $C = \max_{j=1,\dots,m}(\frac{1}{s_j} \times \sum_{i=1}^n p_i)$ will be an upper bound on the completion time of any job in the case of uniform machines. The upper bound reduces to $C = \sum_{i=1}^n p_i$ for identical parallel machines. Thus, we have proved the following corollary.

Corollary 1 *There is a pseudo-polynomial time algorithm that solves the $Pm|DIF|\sum_{i=1}^n(\alpha\max(0, d_i - A) + \beta E_i + \gamma T_i)$ and $Qm|DIF|\sum_{i=1}^n(\alpha\max(0, d_i - A) + \beta E_i + \gamma T_i)$ problems in*

$$O\left(mn\left(\sum_{i=1}^n p_i\right)^{m-1}\right)$$

and

$$O\left(mn\left(\max_{j=1,\dots,m}\left(\frac{1}{s_j} \times \sum_{i=1}^n p_i\right)\right)^{m-1}\right)$$

time, respectively.

2.3 Approximability and approximation on parallel machines

In light of Corollary 1, it is natural to ask whether there exists an FPTAS for the above two problems. As the following theorem shows, however, even the existence of a constant-factor, polynomial time approximation algorithm is extremely unlikely.

Theorem 2 *There is no polynomial time ρ-approximation algorithm for the problem $Pm|DIF|\sum_{i=1}^n(\alpha\max(0, d_i - A) + \beta E_i + \gamma T_i)$ with $\rho < \infty$, unless $\mathcal{P} = \mathcal{NP}$.*

Proof Kovalyov and Werner (2002) have shown recently that the existence of a polynomial time ρ-approximation algorithm for $Pm|d_i = A|\sum_{i=1}^n T_i$ would imply the polynomial solvability of the problem MULTIPROCESSOR SCHEDULING (Garey and Johnson 1979). Thus, assuming $\mathcal{P} \ne \mathcal{NP}$, no such algorithm can exist. The same statement then follows for $Pm|DIF|\sum_{i=1}^n(\alpha\max(0, d_i - A) + \beta E_i + \gamma T_i)$ from its equivalence to $Pm|d_i = A|\sum_{i=1}^n T_i$, which was proved in Theorem 1. □

Since the $Pm|DIF|\sum_{i=1}^n(\alpha\max(0, d_i - A) + \beta E_i + \gamma T_i)$ problem is a special case of the $Qm|DIF|\sum_{i=1}^n(\alpha\max(0, d_i - A) + \beta E_i + \gamma T_i)$ and $Rm|DIF|\sum_{i=1}^n(\alpha\max(0, d_i - A) + \beta E_i + \gamma T_i)$ problems, the above result holds for uniform or unrelated machines, too. One reason the problems are difficult to approximate is that the optimal objective value for an instance may be zero and any algorithm with a guaranteed approximation ratio would have to be optimal for such instances. Furthermore, answering the question whether an instance has a schedule with zero tardiness on parallel machines is NP-hard itself. This provides the motivation for an equivalent formulation of the problem, which has no solution with zero value. This can be done by adding some positive $b > 0$ to the objective function. Although approximating such a version of the problem may be somewhat easier, the next result shows that even this version is unlikely to have an FPTAS.

Theorem 3 *There is no polynomial time ε-approximation algorithm for the problem $Pm|DIF|\sum_{i=1}^n(\alpha\max(0, d_i - A) + \beta E_i + \gamma T_i) + b$ with $\varepsilon < 1/b$, unless $\mathcal{P} = \mathcal{NP}$.*

Proof Kovalyov and Werner (2002) have also shown that the existence of a polynomial time ε-approximation algorithm for $Pm|d_i = A|\sum_{i=1}^n T_i + b$ with $\varepsilon < 1/b$ would also imply the polynomial solvability of the problem MULTIPROCESSOR SCHEDULING (Garey and Johnson 1979). The theorem follows then for the problem $Pm|DIF|\sum_{i=1}^n(\alpha\max(0, d_i - A) + \beta E_i + \gamma T_i) + b$ from its equivalence to $Pm|d_i = A|\sum_{i=1}^n T_i + b$, which was proved in Theorem 1. □

Theorem 3 implies that the case, when the additive term b is some polynomial function of the *size* of the data, may be of interest for approximability. Kolliopoulos and Steiner (2007) have presented an efficient method for obtaining approximation results for the $\sum_{i=1}^n w_i T_i + \sum_{i=1}^n w_i d_i$ objective in various machine environments with given due dates d_i. These approximation results are based on exploiting the close relationship between the $\sum_{i=1}^n w_i(T_i + d_i)$ and the $\sum_{i=1}^n w_i C_i$ objectives:

Theorem 4 (Kolliopoulos and Steiner 2007) *Consider a member $X_0|Y_0|\sum_{i=1}^n w_i C_i$ of the family of non-preemptive scheduling problems $X|Y|\sum_{i=1}^n w_i C_i$ for which there exists a ρ-approximation algorithm. The same algorithm achieves a $(\rho+1)$-approximation for the $X_0|Y_0|\sum_{i=1}^n w_i(T_i + d_i)$ problem.*

Corollary 2 *There is a 2-approximation algorithm with $O(n^3)$ time complexity for the $Rm|DIF|\sum_{i=1}^n(\alpha \max(0, d_i - A) + \beta E_i + \gamma T_i + wA)$ problem.*

Proof It is well-known (see Horn 1973 and Bruno et al. 1974) that the $Rm||w\sum_{i=1}^n C_i$ problem can be solved to optimum in $O(n^3)$ time. According to the above theorem, the optimal schedule for the $Rm||w\sum_{i=1}^n C_i$ problem provides a 2-approximation for the $Rm||w\sum_{i=1}^n (T_i + wA)$ problem, which is equivalent to the problem $Rm|DIF|\sum_{i=1}^n(\alpha \max(0, d_i - A) + \beta E_i + \gamma T_i + wA)$ by Theorem 1. □

2.4 Polynomially solvable cases

In the following we briefly discuss special cases of the $X|DIF|\sum_{i=1}^n(\alpha \max(0, d_i - A) + \beta E_i + \gamma T_i)$ problem which can be solved in polynomial time.

The first one is the case where $X = 1$, i.e., the single-machine problem. For $X = 1$, Seidmann et al. (1981) presented an $O(n \log n)$ optimization algorithm to solve $1|DIF|\sum_{i=1}^n(\alpha \max(0, d_i - A) + \beta E_i + \gamma T_i)$.

The second case is when $A = 0$. This assumption is used in due date assignment problems with no acceptable lead time (e.g., Bagchi et al. 1986a, 1986b; Panwalkar and Rajagopalan 1992; Kahlbacher and Cheng 1993; Chen 1996; Mosheiov 2001; Birman and Mosheiov 2004). This is reasonable when the costumer wants a delivery of an order as soon as possible and may even agree to pay for speedier delivery. When $A = 0$, the $X|DIF|\sum_{i=1}^n(\alpha d_i + \beta E_i + \gamma T_i)$ problem is equivalent to a corresponding $X|d_i = 0|\sum_{i=1}^n T_i$ problem by Theorem 1. This latter problem, of course, is the $X||\sum_{i=1}^n C_i$ problem. The $X||\sum_{i=1}^n C_i$ problem can be solved in $O(n \log n)$ time for $X = Pm$ (see Conway et al. 1967) and in $O(n^3)$ time for $X = Qm$ and $X = Rm$ by solving a linear assignment problem (see Horn 1973 and Bruno et al. 1974). Thus, we have the following corollary.

Corollary 3 *The $X|DIF|\sum_{i=1}^n(\alpha d_i + \beta E_i + \gamma T_i)$ problem can be solved in $O(n \log n)$ time for $X = Pm$ and in $O(n^3)$ time for $X = Qm$ and $X = Rm$.*

Remark 1 Note that the $Pm|CON|\sum_{i=1}^n(\alpha d_i + \beta E_i + \gamma T_i)$ problem is $\mathcal{NP}$-hard (see Cheng and Chen 1994 and De et al. 1994). The last corollary demonstrates that it is the common due date constraint $d_i = d$ for $i = 1, \ldots, n$ that makes that problem hard.

3 The $X|DIF|\sum_{i=1}^n(\alpha \max(0, d_i - A) + \beta U_i)$ problem

In this section our objective is to analyze the $X|DIF|\sum_{i=1}^n(\alpha \max(0, d_i - A) + \beta U_i)$ problem for $X \in \{Pm, Qm, Rm, Fm, Jm, Om\}$. For given exogenous due dates, the $P2||\sum_{i=1}^n U_i$ problem is known to be $\mathcal{NP}$-hard since for $m = 2$, the problem of finding a schedule with value $\sum_{i=1}^n U_i = 0$ for the instance where $A = 1/2 \times \sum_{i=1}^n p_i$ is equivalent to the $\mathcal{NP}$-complete problem MULTIPROCESSOR SCHEDULING (see Garey and Johnson 1979). The $F2||\sum_{i=1}^n U_i$ (and, therefore, also the $J2||\sum_{i=1}^n U_i$) problem is strongly $\mathcal{NP}$-hard even for the case of identical due dates, i.e., $d_i = d$ is fixed for $i = 1, \ldots, n$ (see Lenstra et al. 1977). We will prove that $X|DIF|\sum_{i=1}^n(\alpha \max(0, d_i - A) + \beta U_i)$ is $\mathcal{NP}$-hard, but it is polynomially solvable in some important cases. We will also provide approximation and non-approximability results for the problem on parallel machines.

3.1 Analysis and complexity

For any given schedule, which fixes the set of completion times $\mathbf{C} = (C_1, C_2, \ldots, C_n)$, the problem is reduced to a due date assignment problem, that is to determine the set of due dates, $\mathbf{d} = (d_1, d_2, \ldots, d_n)$ which minimizes the objective given by (2). It is easy to see from (2) that this due date assignment problem has a separable objective function and we can determine the optimal due date for job i by finding d_i that minimizes the following objective for $i = 1, \ldots, n$

$$Z_i(d_i) = \alpha \max(0, d_i - A) + \beta U_i. \tag{9}$$

This due date assignment problem was analyzed by Shabtay and Steiner (2005) in the single-machine case. However, *if* the completion times are fixed, the problem no longer depends on the machine environment. Thus, the same arguments can be used as in the single-machine case, and, therefore, we present the results of this analysis in the following lemma without a proof.

Lemma 3 *For any of the multi-machine scheduling problems $X|DIF|\sum_{i=1}^n(\alpha \max(0, d_i - A) + \beta U_i)$ with $X \in \{Pm, Qm, Rm, Fm, Jm, Om\}$ and a fixed set of completion times $\mathbf{C} = (C_1, C_2, \ldots, C_n)$, the optimal due date assignment policy $\mathbf{d}^* = (d_1^*, \ldots, d_n^*)$ is*

$$d_i^* = \left\{ \begin{array}{ll} \text{any value in } [C_i, A], & \text{if } C_i \le A \\ C_i, & \text{if } A \le C_i \le A + \beta/\alpha \\ \text{any value in } [0, A], & \text{if } C_i \ge A + \beta/\alpha \end{array} \right\}, \quad \text{for } i = 1, \ldots, n. \tag{10}$$

By Lemma 3, under an optimal due date assignment strategy, we have $Z_i = 0$ for any job i with $C_i \le A$; $Z_i = \alpha \times (C_i - A)$ for any job i with $A \le C_i \le A + \beta/\alpha$, and $Z_i = \beta$ for any job i with $C_i > A + \beta/\alpha$. The last case is the only one, where the job will be tardy. As a result, under an optimal due date assignment strategy, our objective becomes

$$\sum_{i=1}^{n} f(C_i), \tag{11}$$

where

$$f(C_i) = \left\{ \begin{array}{ll} \alpha \times \max(0, C_i - A), & \text{if } C_i \le A + \beta/\alpha \\ \beta, & \text{if } C_i \ge A + \beta/\alpha \end{array} \right\}. \tag{12}$$

Theorem 5 *The* $X|DIF|\sum_{i=1}^n(\alpha\max(0,d_i - A) + \beta U_i)$ *problem is equivalent to a corresponding* $X|d_i = A|\sum_{i=1}^n \min(\alpha T_i, \beta)$ *problem with fixed common due date* A *for* $X \in \{Pm, Qm, Rm, Fm, Jm, Om\}$. *Furthermore, the* $X|DIF|\sum_{i=1}^n(\alpha\max(0,d_i - A) + \beta U_i)$ *problem is* $\mathcal{NP}$*-hard for* $X \in \{Pm, Qm, Rm\}$ *and it is strongly* $\mathcal{NP}$*-hard for* $X \in \{Fm, Jm, Om\}$ *even if* $A = 0$.

Proof It is clear from (10–12) that under an optimal due date assignment strategy, $X|DIF|\sum_{i=1}^n(\alpha\max(0,d_i - A) + \beta U_i)$ is equivalent to a corresponding $X|d_i = A|\sum_{i=1}^n \min(\alpha T_i, \beta)$ problem with implied common and non-assignable due date $d_i = A$ for $i = 1, \dots, n$. We will refer to this last objective as the *truncated tardiness*.

For a given set of completion times $\mathbf{C} = (C_1, \dots, C_n)$, define $E = \{i \in J | C_i \le A + \beta/\alpha\}$ as the set of *early jobs* and $T = J \backslash E$ as the set of *tardy jobs*, where $J = \{1, \dots, n\}$ is the set of all jobs. Let us consider an instance of the $X|d_i = A|\sum_{i=1}^n \min(\alpha T_i, \beta)$ problem with a relatively big β value such that $\beta > \alpha \times \max C_i$ for any feasible schedule. (For example, it is sufficient to assume that $\beta > \alpha \times \sum_{i=1}^n p_i$ for $X = Pm$.) For such an instance, the problem becomes an $X|d_i = A|\alpha\sum_{i=1}^n T_i$ problem and no job will be tardy. The $Pm|d_i = A|\alpha\sum_{i=1}^n T_i$ problem is known to be $\mathcal{NP}$-hard for $m = 2$, since the problem of finding a schedule with value $\sum_{i=1}^n T_i = 0$ for the instance where $A = 1/2 \times \sum_{i=1}^n p_i$ is equivalent to the $\mathcal{NP}$-complete problem MULTIPROCESSOR SCHEDULING (see Garey and Johnson 1979). Since the $Pm|d_i = A|\alpha\sum_{i=1}^n T_i$ is a special case of the $Qm|d_i = A|\alpha\sum_{i=1}^n T_i$ and $Rm|d_i = A|\alpha\sum_{i=1}^n T_i$ problems, it is straightforward to see that the last two problems are also $\mathcal{NP}$-hard.

When $A = 0$, the $X|d_i = 0|\sum_{i=1}^n T_i$ problem is equivalent to a $X||\sum_{i=1}^n C_i$ problem for any $X \in \{Pm, Qm, Rm, Fm, Jm, Om\}$. Furthermore, the $X||\sum_{i=1}^n C_i$ problem is known to be strongly $\mathcal{NP}$-hard for $X \in \{F2, J2\}$ (see Garey et al. 1976) and also for $X = O2$ (see Achugbue and Chin 1982). □

3.2 A pseudo-polynomial algorithm for identical and uniform parallel machines

The following lemma gives a useful property of optimal schedules on parallel machines.

Lemma 4 *There exists an optimal schedule for* $Rm|DIF|\sum_{i=1}^n(\alpha\max(0,d_i - A) + \beta U_i)$, *in which the jobs are sequenced according to the SPT rule on each machine.*

Proof Any given job assignment to machines also fixes the actual job processing times. In this case, under an optimal due date assignment strategy, the problem reduces to m unrelated $1||\sum_{i=1}^n f(C_i)$ problems, where $f(C_i)$ is given by (12). In the following we will show that the SPT rule solves each of these problems.

Let us consider an optimal schedule S for one of the $1||\sum_{i=1}^n f(C_i)$ problems and assume that S does not follow the SPT order. Then there are two adjacent jobs q and r in S with $p_q > p_r$, and job q is sequenced before job r. Exchange jobs q and r in the sequence and let the resulting schedule be $\widetilde{S}$. The difference between the objective values is $Z(S) - Z(\widetilde{S}) = f(P_A + p_q) + f(P_A + p_q + p_r) - f(P_A + p_r) - f(P_A + p_r + p_q) = f(P_A + p_q) - f(P_A + p_r)$, where P_A is the sum of the processing times of the jobs sequenced before jobs q and r. It is easy to see that this value is not negative, since $p_q > p_r$ and f is a non-decreasing function. Repeatedly using this exchange argument leads to an optimal SPT schedule on each machine. □

Next we show that the dynamic programming algorithm of Rothkopf (1966) and Lawler and Moore (1969) (described in Sect. 2.2) can be adapted to solve the $Qm|d_i = A|\sum_{i=1}^n \min(\alpha T_i, \beta)$ problem that our $Qm|DIF|\sum_{i=1}^n(\alpha \times \max(0, d_i - A) + \beta U_i)$ problem is equivalent to by Theorem 5.

Since $\sum_{i=1}^n \min(\alpha T_i, \beta)$ is a *regular* criterion, the first condition for the optimality of the algorithm is satisfied. In addition, we know from Lemma 4 that there exists an optimal schedule where the jobs are in SPT order on each machine. The SPT order yields the *same* job sequence on every uniform machine. Therefore, it is possible to index the jobs according to this common SPT order and the jobs assigned to a given machine will be processed in order of their indices. Thus, the optimality conditions of the dynamic programming algorithm are satisfied, and we can apply (5–7) to solve the $Qm|d_i = A|\sum_{i=1}^n \min(\alpha T_i, \beta)$ problem with

$$f_i(t_j) = \left\{ \begin{array}{ll} 0, & \text{if } t_j \le A \\ \min(\alpha \times (t_j - A), \beta), & \text{if } t_j \ge A \end{array} \right\}. \tag{13}$$

For the case of uniform machines, we can use $C = \max_{j=1,\dots,m}(\frac{1}{s_j} \times \sum_{i=1}^n p_i)$ for an upper bound of job completion times, and for the case of identical parallel machines, we can use $C=\sum_{i=1}^n p_i$. Thus, we have proved the following corollary.

Corollary 4 *There is an algorithm that solves the* $Qm|DIF|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)$ *and* $Pm|DIF|\sum_{i=1}^n(\alpha\times\max(0,d_i-A)+\beta U_i)$ *problem in*

$$O\left(mn\left(\max_{j=1,\dots,m}\left(\frac{1}{s_j}\times\sum_{i=1}^n p_i\right)\right)^{m-1}\right)$$

and

$$O\left(mn\left(\sum_{i=1}^n p_i\right)^{m-1}\right)$$

time, respectively.

3.3 Approximability and approximation on parallel machines

In this subsection we look at the issues of approximation for $X|DIF|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)$ on parallel machines.

Theorem 6 *There is no polynomial time* ρ*-approximation algorithm for the problem* $Pm|DIF|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)$ *with* $\rho<\infty$, *unless* $\mathcal{P}=\mathcal{NP}$.

Proof We have proved in Theorem 5 that $Pm|DIF|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)$ is equivalent to $Pm|d_i=A|\sum_{i=1}^n\min(\alpha T_i,\beta)$ with given common due date A. Furthermore, an instance of $Pm|d_i=A|\sum_{i=1}^n\min(\alpha T_i,\beta)$ with sufficiently large β was shown to be equivalent to $Pm|d_i=A|\alpha\sum_{i=1}^n T_i$. The proof then can be completed by the same argument that was used in the proof of Theorem 2. □

Since $Pm|DIF|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)$ is a special case of $Qm|DIF|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)$ and $Rm|DIF|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)$, the above theorem implies the non-approximability of these problems, too. The difficulty may be attributed to the same reasons that we have discussed in the previous section for the $Z=\alpha\max(0,d_i-A)+\beta E_i+\gamma T_i$ objective. Thus, we consider the problem with some $b>0$ added to the objective. As the following theorem shows, in spite of the pseudo-polynomial algorithm of the preceding subsection, the existence of an FPTAS is extremely unlikely for $Pm|DIF|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)+b$ too.

Theorem 7 *There is no polynomial time* ε*-approximation algorithm for the problem* $Pm|DIF|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)+b$ *with* $\varepsilon<1/b$, *unless* $\mathcal{P}=\mathcal{NP}$.

Proof The proof is based on the same argument that was used for proving Theorem 3. □

The theorem implies that it is unlikely that there would be a polynomial time ε-approximation algorithm for $Pm|DIF|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)+b$ if b is bounded by a polynomial in the length of the problem instance (using binary encoding), i.e., by a polynomial in n, $\log\max p_i$, $\log A$, $\log\alpha$, and $\log\beta$. Thus, the case when the additive term b is some polynomial function of the *size* of the data may be of interest for approximability. In the following we show that the method of Kolliopoulos and Steiner (2007) can be extended for approximating the $\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)+b$ objective, too. We use the equivalence of $Pm|DIF|\sum_{i=1}^n(\alpha\max(0,d_i-A)+\beta U_i)+b$ with $Pm|d_i=A|\sum_{i=1}^n\min(\alpha T_i,\beta)+b$ proven in Theorem 5 and $b=n\alpha A$.

Theorem 8 *Consider a member* $X_0|Y_0|\sum_{i=1}^n\min(\alpha C_i,\beta+\alpha A)$ *of the family of non-preemptive scheduling problems* $X|Y|\sum_{i=1}^n\min(\alpha C_i,\beta+\alpha A)$, *for which there exists a* ρ*-approximation algorithm. The same algorithm achieves a* $(\rho+1)$*-approximation for the* $X_0|Y_0,d_i=A|\sum_{i=1}^n(\min(\alpha T_i,\beta)+\alpha A)$ *problem.*

Proof For any schedule σ, let $\mathcal{E}_1(\sigma)=\{i\in J|\ C_i(\sigma)\le A\}$, $\mathcal{E}_2(\sigma)=\{i\in J|\ A\le C_i(\sigma)\le A+\beta/\alpha\}$ and $\mathcal{T}(\sigma)=\{i\in J|C_i(\sigma)\ge A+\beta/\alpha\}$, where $J=\{1,\dots,n\}$ is the set of all jobs. For any σ, we have the following:

$$\sum_{i=1}^n\bigl(\min(\alpha T_i(\sigma),\beta)+\alpha A\bigr)=\sum_{i\in\mathcal{E}_1(\sigma)}\alpha A+\sum_{i\in\mathcal{E}_2(\sigma)}\alpha C_i(\sigma)+\sum_{i\in\mathcal{T}(\sigma)}(\alpha A+\beta)\tag{14}$$

and

$$\sum_{i=1}^n\min(\alpha C_i(\sigma),\beta+\alpha A)\le\sum_{i\in\mathcal{E}_1(\sigma)}\alpha A+\sum_{i\in\mathcal{E}_2(\sigma)}\alpha C_i(\sigma)+\sum_{i\in\mathcal{T}(\sigma)}(\alpha A+\beta).\tag{15}$$

Therefore, we have

$$\sum_{i=1}^n\min\bigl(\alpha C_i(\sigma),\beta+\alpha A\bigr)\le\sum_{i=1}^n\bigl(\min(\alpha T_i(\sigma),\beta)+\alpha A\bigr).\tag{16}$$

Consider now a particular schedule σ^ρ that achieves a ρ-approximation for the objective function $\sum_{i=1}^n\min(\alpha C_i(\sigma),\beta+\alpha A)$, and let σ_C^* and σ_T^* be optimal schedules for the $\sum_{i=1}^n\min(\alpha C_i(\sigma),\beta+\alpha A)$ and the $\sum_{i=1}^n(\min(\alpha T_i,\beta)+\alpha A)$ objectives, respectively. Then

$$\sum_{i=1}^{n}\min(\alpha C_i(\sigma^\rho),\beta+\alpha A)$$
$$\le\rho\sum_{i=1}^{n}\min(\alpha C_i(\sigma_C^*),\beta+\alpha A)$$
$$\le\rho\sum_{i=1}^{n}(\min(\alpha C_i(\sigma_T^*)),\beta+\alpha A)$$
$$\le\rho\sum_{i=1}^{n}(\min(\alpha T_i(\sigma_T^*),\beta)+\alpha A)=\rho OPT, \quad (17)$$

where the second inequality follows from the optimality of σ_C^* for the $\sum_{i=1}^n \min(\alpha C_i, \beta+\alpha A)$ objective, the third inequality is obtained using (16), and by OPT we denote the optimum value for the objective function $\sum_{i=1}^n(\min(\alpha T_i,\beta)+\alpha A)$. From (17) we get that

$$\sum_{i\in\mathcal{E}_2(\sigma^\rho)\cup\mathcal{T}(\sigma^\rho)}\min(\alpha C_i(\sigma^\rho),\beta+\alpha A)$$
$$\le\rho OPT-\sum_{i\in\mathcal{E}_1(\sigma^\rho)}\min(\alpha C_i(\sigma^\rho),\beta+\alpha A), \quad (18)$$

and, using (14), we obtain

$$\sum_{i=1}^{n}(\min(\alpha T_i(\sigma^\rho),\beta)+\alpha A)$$
$$=\sum_{i\in\mathcal{E}_1(\sigma^\rho)}\alpha A+\sum_{i\in\mathcal{E}_2(\sigma^\rho)\cup\mathcal{T}(\sigma^\rho)}\min(\alpha C_i(\sigma^\rho),\beta+\alpha A)$$
$$\le\sum_{i\in\mathcal{E}_1(\sigma^\rho)}\alpha A+\rho OPT$$
$$-\sum_{i\in\mathcal{E}_1(\sigma^\rho)}\min(\alpha C_i(\sigma^\rho),\beta+\alpha A)$$
$$\le(\rho+1)OPT.$$

□

Horn (1973) and Bruno et al. (1974) have solved the $Rm||\sum_{i=1}^n C_i$ problem by formulating it as a linear assignment problem. In the following we show that this approach can also be extended to any truncated objective $\sum_{i=1}^n \min(\alpha C_i,\mu)$ for solving the $Rm||\sum_{i=1}^n \min(\alpha C_i,\mu)$ problem in polynomial time. Let us define l_j as the (unknown) number of 'early' jobs which contribute $\alpha C_i<\mu$ to the total cost when assigned to machine j (for $j=1,\dots,m$) in a schedule. We will say that a fixed value l_j *designates* l_j jobs as early.

Lemma 5 *For a* fixed $\mathbf{l}=(l_1,l_2,\dots,l_m)$, *the schedule minimizing* $\sum_{i=1}^n \min(\alpha C_i,\mu)$ *for the* $Rm||\sum_{i=1}^n \min(\alpha C_i,\mu)$ *problem can be determined by solving a linear assignment problem, which requires* $O((nm)^3)$ *time.*

Proof l_j represents the number of jobs on machine j whose contribution to the total cost is proportional to their completion times, i.e., jobs with $\alpha C_i\le\mu$. It is easy to see that the kth job ($k\le l_j$) among these will contribute its processing time p_i/s_{ij} to its own completion time and the completion time of the (l_j-k) other jobs immediately following it and designated as 'early'. Thus, the cost of assigning job i to the kth position on machine j can be expressed as

$$c_{ijk}(l_j)=\left\{\begin{matrix}\alpha\times(l_j-k+1)\times p_i/s_{ij}, & \text{if } k\le l_j\\ \mu, & \text{if } k>l_j\end{matrix}\right\}.$$
$$\text{for } i,k=1,\dots,n \text{ and } j=1,\dots,m. \quad (19)$$

Let $x_{ijk}=1$ if job i is scheduled in the kth position on machine j and $x_{ijk}=0$, otherwise. Then for any fixed $\mathbf{l}=(l_1,l_2,\dots,l_m)$, our problem can be formulated as follows

$$(P1(\mathbf{l}))\quad \min\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{n}c_{ijk}(l_j)x_{ijk},$$
$$\text{s.t.}\quad \sum_{j=1}^{m}\sum_{k=1}^{n}x_{ijk}=1,\quad \forall i=1,\dots,n;$$
$$\sum_{i=1}^{n}x_{ijk}\le 1,\quad \forall j=1,\dots,m \text{ and } k=1,\dots,n;$$
$$x_{ijk}=0 \text{ or } 1,\quad \forall i,k=1,\dots,n \text{ and } j=1,\dots,m.$$

P1($\mathbf{l}$) is a linear assignment problem with the n jobs on one side and the nm potential positions on the other side. It is well known that such a linear assignment problem can be solved in $O((nm)^3)$ time (see Papadimitriu and Steiglitz 1982). □

The following crucial lemma provides the insight that validates our assignment-based approach for solving $Rm||\sum_{i=1}^n \min(\alpha C_i,\mu)$.

Lemma 6 *Let* $\mathbf{l}^*=(l_1^*,l_2^*,\dots,l_m^*)$ *be the number of jobs designated as early in the schedule* $\sigma(\mathbf{l}^*)$ *that corresponds to the optimal solution of the linear assignment problem* P1($\mathbf{l}^*$) *with the overall smallest cost when we solve the sequence of problems* P1($\mathbf{l}$) *for every one of the possible* $O(n^m)$ *settings of* $\mathbf{l}$. *Then every job designated as early in* $\sigma(\mathbf{l}^*)$ *will be completed before* μ/α.

Proof Let us denote by $E(\sigma(\mathbf{l}^*))$ the jobs designated as early, i.e., the first l_j^* jobs scheduled on machine j by *P1*($\mathbf{l}^*$) for $j=1,\dots,m$. Suppose that, contrary to the lemma, there is a job $k\in E(\sigma(\mathbf{l}^*))$, which is scheduled on machine j and has a completion time $C_k>\mu/\alpha$, and without loss of generality, let k be the one with the largest completion time among the jobs with this property. Then the cost of this job in *P1*($\mathbf{l}^*$)

was $\alpha C_k > \mu$. Consider, however, the minimum cost assignment for *P1*($\bar{\mathbf{l}}$) with $\bar{\mathbf{l}} = (l_1^*, \ldots, l_{j-1}^*, l_j^* - 1, l_{j+1}^*, \ldots, l_m^*)$. Notice that the assignment which designates exactly the jobs in $E(\sigma(\mathbf{l}^*))\backslash\{k\}$ as early is a feasible solution for *P1*($\bar{\mathbf{l}}$), and job k has cost μ in this solution, while every other job has the same cost as in the optimal solution for *P1*($\mathbf{l}^*$). Thus, *P1*($\bar{\mathbf{l}}$) has a solution whose cost is strictly less than the optimal cost of *P1*($\mathbf{l}^*$). This, however, contradicts the definition of *P1*($\mathbf{l}^*$). □

Theorem 9 *The $Rm||\sum_{i=1}^n \min(\alpha C_i, \mu)$ problem can be solved in $O(n^{m+3}m^3)$ time by solving a sequence of linear assignment problems.*

Proof Lemma 6 ensures that all jobs $i \in E(\sigma(\mathbf{l}^*))$ will indeed contribute αC_i to the objective in $P1(\mathbf{l}^*)$. Then the theorem immediately follows from the preceding two lemmas, observing that there are at most $O(n^m)$ possible settings for the $\mathbf{l}$ values. □

Corollary 5 *There is an algorithm, which in $O(n^{m+3}m^3)$ time finds a schedule that is a 2-approximation for $Rm|DIF|\sum_{i=1}^n(\alpha \max(0, d_i - A) + \beta U_i + \alpha A)$.*

Proof By Theorem 5, solving the $Rm|DIF|\sum_{i=1}^n(\alpha \max(0, d_i - A) + \beta U_i + \alpha A)$ problem is equivalent to solving $Rm|d_i = A|\sum_{i=1}^n(\min(\alpha T_i, \beta) + \alpha A)$. By Theorem 8, any algorithm that solves $Rm||\sum_{i=1}^n \min(\alpha C_i, \beta + \alpha A)$ to optimum also yields a schedule for $Rm|d_i = A|\sum_{i=1}^n(\min(\alpha T_i, \beta) + \alpha A)$ that is a 2-approximation. Applying Theorem 9 with $\mu = \beta + \alpha A$ gives such an algorithm. □

3.4 Polynomially solvable cases

In the following we discuss special cases of the $X|DIF|\sum_{i=1}^n(\alpha \max(0, d_i - A) + \beta U_i)$ problem, which can be solved in polynomial time. The first one is the case where $X = 1$, i.e., the single-machine problem. Shabtay and Steiner (2005) presented an $O(n \log n)$ optimization algorithm to solve the $1|DIF|\sum_{i=1}^n(\alpha \max(0, d_i - A) + \beta U_i)$ problem.

The second case is where $A = 0$. The corresponding $X|DIF|\sum_{i=1}^n(\alpha d_i + \beta U_i)$ problem reduces to $X||\sum_{i=1}^n \min(\alpha C_i, \beta)$, by Theorem 5. This problem is still strongly $\mathcal{NP}$-hard for $X \in \{Fm, Jm, Om\}$ for $m \geq 2$ (see Theorem 5). On the other hand, as the next theorem shows, the $Rm|DIF|\sum_{i=1}^n(\alpha d_i + \beta U_i)$ problem can be solved in polynomial time.

Theorem 10 *There is an algorithm that solves the $Rm|DIF|\sum_{i=1}^n(\alpha d_i + \beta U_i)$ problem in $O(n^{m+3})$ time for a fixed number of machines m.*

Proof We showed in Theorem 9 and Lemma 5 that for a given $\mathbf{l}$ the equivalent $Rm||\sum_{i=1}^n \min(\alpha C_i, \beta)$ problem can be solved in $O(n^3)$ time for a fixed m by solving a linear assignment problem. Since we have no more than $O(n^m)$ different $\mathbf{l}$ vectors to consider, the problem can be solved in $O(n^{m+3})$ time. □

Next we present a polynomial time solution with improved complexity for the $Pm|DIF|\sum_{i=1}^n(\alpha d_i + \beta U_i)$ problem, by again solving the equivalent $Pm||\sum_{i=1}^n \min(\alpha C_i, \beta)$ problem.

Lemma 7 *If $C_k \geq \beta/\alpha$ and $C_h \leq \beta/\alpha$ in an optimal schedule for the $Pm||\sum_{i=1}^n \min(\alpha C_i, \beta)$ problem then $p_h \leq p_k$.*

Proof Let us assume that there is an optimal schedule S with $C_k \geq \beta/\alpha$, $C_h \leq \beta/\alpha$ and $p_h > p_k$. By Lemma 4, we can assume that jobs k and h are processed on different machines. Let job h be processed on machine q and job k on machine r. Interchange the two jobs and let the new schedule be $\widetilde{S}$. Let A_q be the set of jobs processed before job h on machine q in schedule S, A_r the set of jobs processed before job k on machine r, B_q the set of jobs processed after job h on machine q and B_r the set of jobs processed after k on machine r.

We prove first that the total cost of jobs on machine r is the same in S and $\widetilde{S}$. Since the job sequence and completion times for set A_r are identical in S and $\widetilde{S}$, the total costs of this set in the two schedules are the same. Since $C_k(S) \geq \beta/\alpha$, we also have $C_h(\widetilde{S}) = C_k(S) + p_h - p_k > \beta/\alpha$. Therefore, all jobs sequenced after set A_r are tardy in both schedules, and each of those jobs has a cost β independent of their processing sequence.

We prove next that the total cost of jobs on machine q in schedule $\widetilde{S}$ is less than in schedule S. Since the job sequence and completion times for set A_q are identical in S and $\widetilde{S}$, the total cost of this set is the same in both schedules. Since $C_k(\widetilde{S}) = C_h(S) - p_h + p_k < C_h(S) \leq \beta/\alpha$, the cost of job k in schedule $\widetilde{S}$ is less than the cost of job h in schedule S. In addition, it is easy to observe that for any $j \in B_q$ we have $C_j(\widetilde{S}) = C_j(S) - p_h + p_k < C_j(S)$. Since the cost is a non-decreasing function of the completion time, the total cost of set B_q under schedule $\widetilde{S}$ is not greater than its total cost under schedule S. Therefore, the total cost for the jobs scheduled on machine q under schedule $\widetilde{S}$ is less than under schedule S. This contradicts the optimality of S and completes our proof. □

As a consequence of Lemma 7, we will show that the following algorithm of Conway et al. (1967) that solves the $Pm||\sum_{i=1}^n C_i$ problem in $O(n \log n)$ time also solves the $Pm||\sum_{i=1}^n \min(\alpha C_i, \beta)$ problem.

Algorithm 1 (Conway et al. 1967) Optimization algorithm for $Pm||\sum_{i=1}^n C_i$.

Step 1. Reindex the jobs in a non-decreasing order of processing times (in SPT order).

Step 2. Assign job i to machine $i - \lfloor \frac{i}{m} \rfloor \times m$ for $i = 1, \dots, n$.

It is straightforward to show for the schedule produced by the above algorithm that if $C_h \leq C_k$ then $p_h \leq p_k$.

Lemma 8 *Algorithm 1 solves the* $Pm||\sum_{i=1}^{n} \min(\alpha C_i, \beta)$ *problem.*

Proof Early jobs, i.e., jobs whose completion time is not greater than β/α, can be scheduled according to Algorithm 1, since for this set the objective becomes to minimize $\sum_{i=1}^{n} C_i$. Since all tardy jobs have the same cost β, the sequence and machine assignment within this set is immaterial, i.e., they can also be sequenced according to Algorithm 1. The proof is completed by the result given in Lemma 7, i.e., that each early job has a processing time not greater than that of any tardy job. □

Corollary 6 *The* $Pm|DIF|\sum_{i=1}^{n}(\alpha d_i + \beta U_i)$ *problem can be solved in* $O(n \log n)$ *time.*

Proof By Theorem 5, the $Pm|DIF|\sum_{i=1}^{n}(\alpha d_i + \beta U_i)$ problem is equivalent to the $Pm||\sum_{i=1}^{n} \min(\alpha C_i, \beta)$ problem, and the latter problem can be solved in $O(n \log n)$ time by Lemma 8. □

Remark 2 We note that the $Pm|CON|\sum_{i=1}^{n}(\alpha d_i + \beta U_i)$ problem is $\mathcal{NP}$-hard (see Kahlbacher and Cheng 1993). The last corollary demonstrates that it is the common due date constraint $d_i = d$ for $i = 1, \dots, n$ that makes that problem hard.

4 Summary

We have studied two multi-machine scheduling problems with tardiness penalties and due date assignment. In contrast with most of the literature, we assumed that different due dates can be assigned to different jobs. We have shown that both problems are either strongly $\mathcal{NP}$-hard or $\mathcal{NP}$-hard in the ordinary sense, dependent on the machine environment. We also presented polynomial-time solutions for some important special cases. Some of these cases demonstrated that a problem may be $\mathcal{NP}$-hard with common due date assignment, but be polynomially solvable if different individual due dates are allowed. We have also shown that although $Pm||\sum(\beta_i E_i + \gamma_i T_i)$ and $Pm||\sum U_i$ are $\mathcal{NP}$-hard with given due dates, they become polynomially solvable with due date assignment with no acceptable lead time ($A = 0$). This makes the integration of scheduling and due date assignment even more attractive for practical applications.

Acknowledgements The authors would like to thank two anonymous referees and the Associate Editor for their comments, which led to a better presentation for the paper.

References

Achugbue, J. O., & Chin, F. Y. (1982). Scheduling the open shop to minimize mean flow time. *SIAM Journal on Computing*, *11*, 709–720.

Baker, K. R., & Scudder, G. D. (1990). Sequencing with earliness and tardiness penalties: a review. *Operations Research*, *38*, 22–36.

Bagchi, U., Sullivan, R. S., & Chang, Y. L. (1986a). Minimizing mean absolute deviation of completion times about a common due date. *Naval Research Logistics*, *33*, 227–240.

Bagchi, U., Sullivan, R. S., & Chang, Y. L. (1986b). Minimizing mean squared deviation of completion times about a common due date. *Management Science*, *33*, 894–906.

Birman, M., & Mosheiov, G. (2004). A note on due-date assignment in a two-machine flow-shop. *Computers and Operations Research*, *31*(3), 473–480.

Bruno, J. L., Coffman, E. G., & Sethi, R. (1974). Scheduling independent tasks to reduce mean finishing time. *Communications of the ACM*, *17*, 382–387.

Chen, Z. L. (1996). Scheduling and common due date assignment with earliness-tardiness penalties and batch delivery costs. *European Journal of Operational Research*, *93*, 49–60.

Cheng, T. C. E., & Chen, Z. L. (1994). Parallel-machine scheduling problems with earliness and tardiness penalties. *Journal of the Operational Research Society*, *45*, 685–695.

Cheng, T. C. E., & Kovalyov, M. Y. (1996). Batch scheduling and common due-date assignment on a single machine. *Discrete Applied Mathematics*, *70*, 231–245.

Conway, W. L., Maxwell, W. L., & Miller, L. W. (1967). *Theory of scheduling*. Reading: Addison–Wesley.

De, P., Ghosh, J. B., & Wells, C. E. (1991). Optimal delivery time quotation and order sequencing. *Decision Sciences*, *22*, 379–390.

De, P., Ghosh, J. B., & Wells, C. E. (1994). Due-date assignment and early/tardy scheduling on identical parallel machines. *Naval Research Logistics*, *41*, 17–32.

Du, J.Y., & Leung, T. (1990). Minimizing total tardiness on one machine is $\mathcal{NP}$-hard. *Mathematics of Operations Research*, *15*, 483–495.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of $\mathcal{NP}$-completeness*. San Francisco: Freeman.

Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flow shop and job shop scheduling. *Mathematics of Operations Research*, *1*, 117–129.

Gordon, V., Proth, J. M., & Chu, C. B. (2002). A survey of the state-of-the-art of common due date assignment and scheduling research. *European Journal of Operational Research*, *139*, 1–25.

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

Horn, W. A. (1973). Minimizing average flow time with parallel machines. *Operations Research*, *21*, 846–847.

Kahlbacher, H. G., & Cheng, T. C. E. (1993). Parallel machine scheduling to minimize costs for earliness and number of tardy jobs. *Discrete Applied Mathematics*, *47*, 139–164.

Kolliopoulos, S. G., & Steiner, G. (2007, in preparation). Approximation algorithms for scheduling problems with a modified total weighted tardiness objective.

Kovalyov, M., & Werner, F. (2002). Approximation schemes for scheduling jobs with common due date on parallel machines to minimize total tardiness. *Journal of Heuristics*, *8*, 415–428.

Lawler, E. L., & Moore, J. M. (1969). A functional equation and its application to resource allocation and sequencing problems. *Management Science*, *16*, 77–84.

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

Mosheiov, G. (2001). A common due-date assignment problem on parallel identical machines. *Computers and Operations Research*, *28*(8), 719–732.

Panwalkar, S. S., Smith, M. L., & Seidmann, A. (1982). Common Due date assignment to minimize total penalty for the one machine scheduling problem. *Operations Research*, *30*, 391–399.

Panwalkar, S. S., & Rajagopalan, R. (1992). Single-machine sequencing with controllable processing times. *European Journal of Operational Research*, *59*, 298–302.

Papadimitriu, C. H., & Steiglitz, K. (1982). *Combinatorial optimization: algorithms and complexity*. Englewood Cliffs: Prentice-Hall.

Rothkopf, M. H. (1966). Scheduling independent tasks on parallel processors. *Management Science*, *12*, 437–447.

Seidmann, A., Panwalkar, S. S., & Smith, M. L. (1981). Optimal assignment of due dates for a single processor scheduling problem. *International Journal of Production Research*, *19*, 393–399.

Shabtay, D., & Steiner, G. (2005). Two due date assignment problems in scheduling a single machine, *Operations Research Letters*, *34*(6), 683–691.

Slotnick, S. A., & Sobel, M. J. (2005). Manufacturing lead-time rules: customer retention versus tardiness costs. *European Journal of Operational Research*, *169*, 825–856.