RAIRO-Oper. Res. 52 (2018) 359–370
https://doi.org/10.1051/ro/2017074

RAIRO Operations Research
www.rairo-ro.org

BICRITERIA SCHEDULING FOR DUE DATE ASSIGNMENT WITH TOTAL WEIGHTED TARDINESS *

Hao Lin[1], Cheng He[1] and Yixun Lin[2,*]

Abstract. In the due date assignment, the bicriteria scheduling models are motivated by the trade-off between the due date assignment cost and a performance criterion of the scheduling system. The bicriteria scheduling models related to the maximum tardiness and the weighted number of tardy jobs have been studied in the literature. In this paper we consider a new model with criteria of the due date assignment cost and the total weighted tardiness. The main results are polynomial-time algorithms for the linear combination version, the constraint version, and the Pareto optimization version of bicriteria scheduling.

Mathematics Subject Classification. 90B35, 90B50, 90C29.

Received July 4, 2016. Accepted October 5, 2017.

1. Introduction

The due date assignment problems have been studied extensively in the literature, where the main contributions can be consulted in books [2, 3] and surveys [4, 6, 8]. The early research in this area focused on the common due date assignment, in which a due date d common to all jobs is determined, so that the earliness and the tardiness are minimized. This is generally related to the just-in-time (JIT) scheduling. Henceforth, many generalized models with different restrictions on the due dates and with different objective functions were put forward. For instance, the due dates are different, and the release dates may be also considered [5], the due dates are dependent on the processing times [8], the processing times are controllable [15], or resource-dependent [13], or positionally dependent [7]. Recently, the optimal restricted due date assignment was studied systematically in [16].

In most of due date assignment models, there is a trade-off between the due date assignment cost and the performance criterion of the scheduling system. In fact, the customers always demand that the quoted due dates should be met; otherwise, considerable penalties will be applied. On the other hand, the manufacturer may want to improve the system performance by optimizing a cost criterion related to the due dates. Generally speaking, the smaller the due dates are, the lower for the assignment cost; while the greater the due dates are, the better

Keywords. Scheduling, bicriteria scheduling, due date assignment, total weighted tardiness, polynomial algorithm.

* *Supported by NSFC (11201121, 11571323) and NSFSTDOHN (162300410221).*

[1] School of Science, Henan University of Technology, Zhengzhou 450001, China.

[2] School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China.

* Corresponding author: linyixun@zzu.edu.cn

for the performance criterion. Motivated by this kind of benefit balance, the bicriteria scheduling in due date assignment comes up in connection with the recent trend of multicriteria scheduling (see [9,17]).

According to the category of multicriteria optimization, there are three basic versions for a bicriteria problem with criteria f_1 and f_2: (1) the linear combination version, that is, the single objective function is in the form $\alpha f_1 + \beta f_2$; (2) the constraint version, that is, minimizing f_1 subject to $f_2 \leq k$ or conversely; (3) the Pareto optimization version, that is, to identify the set of all Pareto optimal solutions. These are in fact three related optimization problems.

In particular, the bicriteria scheduling problem for due date assignment with weighted number of tardy jobs in Shabtay and Steiner [12] and Koulamas [10] is to minimize

$$TC = \alpha \sum_{j=1}^{n} d_j + \sum_{j=1}^{n} w_j U_j,$$

where d_j and w_j are the due date and the weight, respectively, of job j with $1 \leq j \leq n$ and U_j denotes the tardiness indicator defined by $U_j = 1$ if $C_j > d_j$ and $U_j = 0$ otherwise (where C_j is the completion time of job j). This is indeed the linear combination version of the bicriteria scheduling with criteria $f_1 = \sum_{j=1}^{n} d_j$ and $f_2 = \sum_{j=1}^{n} w_j U_j$, where α is the combination coefficient. For this model, an $O(n^4)$ algorithm and an $O(n^2)$ algorithm have been obtained in [10,12]. Furthermore, [11] studied the constraint version and the Pareto optimization version of this bicriteria scheduling problem, for which the NP-hardness, polynomially solvable cases and polynomial-time approximation scheme were presented.

Moreover, Shabtay *et al.* [14] studied the bicriteria scheduling with due date assignment cost $F(\mathbf{d}) = \sum_{j=1}^{n} f_j(d_j)$ and maximum tardiness $T_{\max}$, where f_j is a non-decreasing function for each j and $T_{\max} = \max_{1 \leq j \leq n} T_j$ with $T_j = \max\{0, C_j - d_j\}$. The topic of this article contains the three versions as follows:

- The linear combination version: to minimize $\alpha F(\mathbf{d}) + \beta T_{\max}$;
- The constraint version: to minimize $T_{\max}$ subject to $F(\mathbf{d}) \leq D$ or to minimize $F(\mathbf{d})$ subject to $T_{\max} \leq T$;
- The Pareto optimization version: to identify the set of Pareto optimal solutions of two criteria $F(\mathbf{d})$ and $T_{\max}$.

Herein, all these versions were shown to be NP-hard even on a single machine; and then polynomial-time algorithms were established for a series of special cases (for example, every job has the same function f_j or has the same processing time p_j). The approximation algorithms were also presented.

In addition to the weighted number of tardy jobs $\sum w_j U_j$ and the maximum tardiness $T_{\max}$ mentioned above, an important due date involving criterion is the total weighted tardiness $\sum w_j T_j$. In this paper we study a new model of single machine bicriteria scheduling problem by taking the total weighted tardiness $\sum w_j T_j$ as the second criterion. In more detail, the two criteria are the due date assignment cost $f_1 = \sum_{j=1}^{n} a_j d_j$, where $a_j > 0$ stands for the cost of one-unit of d_j, and the total weighted tardiness $f_2 = \sum_{j=1}^{n} w_j T_j$. Our goal is to establish polynomial-time algorithms for the three versions of bicriteria scheduling: the linear combination version, the constraint version, and the Pareto optimization version respectively.

The paper is organized as follows. In Section 2 we describe some basic notations and problem formulations. In Section 3 we discuss the linear combination version. Section 4 is concerned with the constraint version. Section 5 is dedicated to the Pareto optimization version. Finally Section 6 contains a summary.

2. Preliminaries

To state more precisely, let us introduce some notations, following the textbooks [2,3]. Let $J_1, J_2, \ldots, J_n$ be n jobs with processing times $p_1, p_2, \ldots, p_n$ and due dates $d_1, d_2, \ldots, d_n$ respectively. For convenience, we may denote the set of jobs by $\{1, 2, \ldots, n\}$. A *schedule* of jobs to be processed on a single machine is defined as a permutation $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$ of $\{1, 2, \ldots, n\}$. For a schedule π, the *completion time* of job $\pi(i)$ is $C_{\pi(i)} = \sum_{j=1}^{i} p_{\pi(j)}$. Then the *tardiness* is $T_{\pi(i)} = \max\{0, C_{\pi(i)} - d_{\pi(i)}\}$. Moreover, the *tardiness indicator* is defined by $U_{\pi(i)} = 1$ if $T_{\pi(i)} > 0$ and $U_{\pi(i)} = 0$ otherwise. In the environment of due date assignment, a *feasible* *solution* consists of a schedule π of jobs and an assignment $\mathbf{d} = (d_{\pi(1)}, d_{\pi(2)}, \ldots, d_{\pi(n)})$ of due dates. So this feasible solution is denoted by $\sigma = (\pi, \mathbf{d})$ and the objective function is denoted by $f(\sigma) = f(\pi, \mathbf{d})$ for some function f.

The single machine bicriteria scheduling models discussed in this paper are as follows.

- The linear combination version, denoted by $1||\alpha f_1 + \beta f_2$, is a single criterion problem with objective function

$$f(\pi, \mathbf{d}) = \alpha \sum_{i=1}^{n} a_{\pi(i)} d_{\pi(i)} + \beta \sum_{i=1}^{n} w_{\pi(i)} T_{\pi(i)}. \tag{2.1}$$

- The constraint version, denoted by $1|f_1 \le D|f_2$, is the following mathematical programming:

$$\min \quad \textstyle\sum_{i=1}^{n} w_{\pi(i)} T_{\pi(i)} \tag{2.2}$$

$$\text{s.t.} \ \textstyle\sum_{i=1}^{n} a_{\pi(i)} d_{\pi(i)} \le D \tag{2.3}$$

$$d_{\pi(i)} \ge 0, \quad 1 \le i \le n. \tag{2.4}$$

- The Pareto optimization version, denoted by $1||(f_1, f_2)$, is to determine the set of Pareto optimal solutions of two criteria $\sum_{i=1}^{n} a_{\pi(i)} d_{\pi(i)}$ and $\sum_{i=1}^{n} w_{\pi(i)} T_{\pi(i)}$.

For the last version, we may address some basic concepts of Pareto (or simultaneous) optimization (see [9, 17] for details).

In a bicriteria optimization problem with performance criteria f_1 and f_2, a feasible solution σ is said to be *Pareto optimal* if there is no feasible solution σ' such that $f_1(\sigma') \le f_1(\sigma)$ and $f_2(\sigma') \le f_2(\sigma)$ where at least one of the inequalities is strict.

In a single machine bicriteria scheduling problem with criteria f_1 and f_2, the simultaneous (Pareto) optimization version $1||(f_1, f_2)$, is to identify the set of Pareto optimal solutions.

For a feasible solution σ, we may associate a point $(f_1(\sigma), f_2(\sigma))$ in $\mathbb{R}^2$. If σ is a Pareto optimal solution, then we say that $(f_1(\sigma), f_2(\sigma))$ is a *Pareto optimal point*. However, a Pareto optimal point may correspond to different Pareto optimal solutions σ having the same objective values $f_1(\sigma)$ and $f_2(\sigma)$ (they are equivalent). When there is no confusion, we may say Pareto optimal solutions and Pareto optimal points interchangeably. The set of the Pareto optimal points may be finite or infinite. In the latter case, a curve containing all Pareto optimal points is called the *trade-off curve* or *efficient frontier*.

3. Linear combination version

We first consider the linear combination version of (2.1), denoted by $1||\alpha \sum a_j d_j + \beta \sum w_j T_j$. Without loss of generalization, we may assume $\beta = 1$. So the problem is to minimize the following single objective function

$$\begin{aligned} f(\pi, \mathbf{d}) &= \alpha \sum_{i=1}^{n} a_{\pi(i)} d_{\pi(i)} + \sum_{i=1}^{n} w_{\pi(i)} T_{\pi(i)} \\ &= \alpha \sum_{i=1}^{n} a_{\pi(i)} d_{\pi(i)} + \sum_{i=1}^{n} w_{\pi(i)} \max\{0, C_{\pi(i)} - d_{\pi(i)}\}. \end{aligned}$$

Lemma 3.1. *For the problem $1||\alpha \sum a_j d_j + \sum w_j T_j$, there exists an optimal solution $\sigma = (\pi, \mathbf{d})$ such that $d_{\pi(i)} \le C_{\pi(i)}$ for $1 \le i \le n$.*

Proof. Let $\sigma = (\pi, \mathbf{d})$ be an optimal solution with $d_{\pi(k)} > C_{\pi(k)}$ for some k. Then by letting $d'_{\pi(k)} = C_{\pi(k)}$ and $d'_{\pi(i)} = d_{\pi(i)}$ for $i \ne k$, we have

$$T(\pi, \mathbf{d}') = \sum_{i=1}^{n} w_{\pi(i)} \max\{0, C_{\pi(i)} - d'_{\pi(i)}\} = \sum_{i=1}^{n} w_{\pi(i)} \max\{0, C_{\pi(i)} - d_{\pi(i)}\} = T(\pi, \mathbf{d}).$$

Moreover, $\sum_{1\le i\le n} a_{\pi(i)}d'_{\pi(i)} < \sum_{1\le i\le n} a_{\pi(i)}d_{\pi(i)}$. Hence $f(\pi,\mathbf{d}') \le f(\pi,\mathbf{d})$ and thus $\sigma' = (\pi,\mathbf{d}')$ is also an optimal solution. □

By this lemma, we may assume that $d_{\pi(i)} \le C_{\pi(i)}$ $(1 \le i \le n)$ for any given schedule π. Thus the objective function can be written as

$$f(\pi,\mathbf{d}) = \alpha\sum_{i=1}^{n} a_{\pi(i)}d_{\pi(i)} + \sum_{i=1}^{n} w_{\pi(i)}\max\{C_{\pi(i)}, d_{\pi(i)}\} - \sum_{i=1}^{n} w_{\pi(i)}d_{\pi(i)}$$
$$= \sum_{i=1}^{n} w_{\pi(i)}C_{\pi(i)} + \sum_{i=1}^{n}(\alpha a_{\pi(i)} - w_{\pi(i)})d_{\pi(i)}.$$

For minimizing the last summation of the above representation, the assignment of due dates can be obtained by taking each $d_{\pi(i)}$ with negative coefficient as large as possible, namely $d_{\pi(i)} = C_{\pi(i)}$, and setting $d_{\pi(i)} = 0$ for the others. Hence

$$d_{\pi(i)} = \begin{cases} C_{\pi(i)}, & \text{if} \quad \alpha a_{\pi(i)} < w_{\pi(i)} \\ 0, & \text{otherwise.} \end{cases}$$

Thus we have

$$f(\pi,\mathbf{d}) = \sum_{i=1}^{n} w_{\pi(i)}C_{\pi(i)} + \sum_{\alpha a_{\pi(i)} < w_{\pi(i)}} (\alpha a_{\pi(i)} - w_{\pi(i)})C_{\pi(i)} = \sum_{i=1}^{n} w'_{\pi(i)}C_{\pi(i)}, \tag{3.1}$$

where $w'_{\pi(i)} = \min\{\alpha a_{\pi(i)}, w_{\pi(i)}\}, 1 \le i \le n$.

To summarize, the problem of minimizing $f(\pi,\mathbf{d})$ is equivalent to the classical problem $1||\sum w_jC_j$ with new weights w'_j. It is well known that this problem can be solved by the WSPT (the weighted shortest processing time first) rule, that is, in the nondecreasing order of the ratios $\frac{p_j}{w_j}$ (see [2,3]). Thus, we have an algorithm of Revised WSPT rule for solving problem $1||\alpha\sum a_jd_j + \sum w_jT_j$ as follows.

Algorithm R-WSPT

(1) Let $w'_j = \min\{\alpha a_j, w_j\}, 1 \le j \le n$.
(2) Determine the schedule π by the WSPT rule such that

$$\frac{p_{\pi(1)}}{w'_{\pi(1)}} \le \frac{p_{\pi(2)}}{w'_{\pi(2)}} \le \cdots \le \frac{p_{\pi(n)}}{w'_{\pi(n)}}.$$

(3) Determine the due date assignment $\mathbf{d}$ by

$$d_{\pi(i)} = \begin{cases} C_{\pi(i)}, & \text{if} \quad \alpha a_{\pi(i)} < w_{\pi(i)} \\ 0, & \text{otherwise.} \end{cases}$$

(4) Return the optimal solution $\sigma = (\pi,\mathbf{d})$.

Theorem 3.2. *Algorithm R-WSPT correctly solves the problem* $1||\alpha\sum a_jd_j + \sum w_jT_j$ *in* $O(n\log n)$ *time.*

Proof. The correctness of the algorithm is based on the representation of (3.1). As for the running time of the algorithm, the new weights w'_j can be computed in Step (1) in $O(n)$ time. Step (2) for determining schedule π by the WSPT rule runs in $O(n\log n)$ time. In Step (3), the due date assignment $\mathbf{d}$ can be computed in $O(n)$ time. So the overall complexity is $O(n\log n)$. □

4. Constraint version

We next consider the constraint version $1|\sum a_j d_j \le D|\sum w_j T_j$, namely problem (2.2)–(2.4). A solution $\sigma = (\pi, \mathbf{d})$ is said to be *feasible* if it satifies the constraint $\sum_{1\le i\le n} a_{\pi(i)} d_{\pi(i)} \le D$. An optimal solution is a feasible solution that minimizes the objective function $T(\pi, \mathbf{d}) = \sum_{1\le i\le n} w_{\pi(i)} T_{\pi(i)}$. As in Lemma 3.1, we have the following observation.

Lemma 4.1. *For problem $1|\sum a_j d_j \le D|\sum w_j T_j$, there exists an optimal solution $\sigma = (\pi, \mathbf{d})$ such that $d_{\pi(i)} \le C_{\pi(i)}$ for $1 \le i \le n$.*

So we may assume $d_{\pi(i)} \le C_{\pi(i)}$ $(1 \le i \le n)$ for any given schedule π. Thus the objective function can be written as

$$T(\pi, \mathbf{d}) = \sum_{i=1}^{n} w_{\pi(i)} \max\{C_{\pi(i)}, d_{\pi(i)}\} - \sum_{i=1}^{n} w_{\pi(i)} d_{\pi(i)} = \sum_{i=1}^{n} w_{\pi(i)} C_{\pi(i)} - \sum_{i=1}^{n} w_{\pi(i)} d_{\pi(i)}.$$

Furthermore, we can write $d_{\pi(i)} = C_{\pi(i)} x_{\pi(i)}$ with $0 \le x_{\pi(i)} \le 1$ $(1 \le i \le n)$ and the feasible solution is denoted by $\sigma = (\pi, \mathbf{x})$. Thus the objective function is represented as

$$T(\pi, \mathbf{x}) = \sum_{i=1}^{n} w_{\pi(i)} C_{\pi(i)} - \sum_{i=1}^{n} w_{\pi(i)} x_{\pi(i)} C_{\pi(i)} = \sum_{i=1}^{n} w_{\pi(i)} (1 - x_{\pi(i)}) C_{\pi(i)}. \tag{4.1}$$

When $D = 0$, and so all $d_j = x_j = 0$, the optimal solution can be obtained by sequencing the jobs in WSPT order. We assume $D > 0$ in the sequel.

Lemma 4.2. *There exists an optimal solution $\sigma = (\pi, \mathbf{x})$ such that:*

(a) *there is at most one job with $0 < x_j < 1$;*
(b) *if the jobs are indexed in the order that $\frac{w_1}{a_1} \le \frac{w_2}{a_2} \le \dots \le \frac{w_n}{a_n}$, then the set of jobs with $x_j > 0$ is $\{k, k+1, \dots, n\}$ for some k.*

Proof. According to equation (4.1), for an optimal schedule π (where $C_{\pi(i)}$ are known), the problem for determining $\mathbf{x}$ is the following Linear Programming:

$$\max\ z(\pi, \mathbf{x}) = \textstyle\sum_{i=1}^{n} w_{\pi(i)} C_{\pi(i)} x_{\pi(i)} \tag{4.2}$$

$$\text{s.t.} \qquad \textstyle\sum_{i=1}^{n} a_{\pi(i)} C_{\pi(i)} x_{\pi(i)} \le D \tag{4.3}$$

$$0 \le x_{\pi(i)} \le 1, \quad 1 \le i \le n. \tag{4.4}$$

In this LP, there exists an optimal solution $\mathbf{x}$ which is a basic feasible solution, that is a vertex of the polytope of feasible region. By the constraints (4.3)−(4.4), we can see that each basic feasible solution $\mathbf{x}$ has some components with $x_j = 1$, some components with $x_j = 0$, and at most one component with $0 < x_j < 1$ corresponding to the equality of (4.3).

Suppose that $\sigma = (\pi, \mathbf{x})$ is optimal. Let $S = \{j : x_j > 0\}$ and $\bar{S} = \{j : x_j = 0\}$. To show (b), it suffices to prove that $\frac{w_i}{a_i} \le \frac{w_j}{a_j}$ for any $i \in \bar{S}$ and $j \in S$. Assume, to the contrary, that $\frac{w_i}{a_i} > \frac{w_j}{a_j}$ for some $i \in \bar{S}$ and $j \in S$. Then we define a new solution $\sigma' = (\pi, \mathbf{x}')$ by setting

$$x'_j = x_j - \varepsilon, \qquad x'_i = x_i + \frac{a_j C_j}{a_i C_i}\varepsilon,$$

where $\varepsilon > 0$ is sufficiently small, C_j is the completion time of job j for $1 \le j \le n$ under the schedule π, and $x'_h = x_h$ for $h \ne i, j$. Then we have

$$a_i C_i x'_i + a_j C_j x'_j = a_i C_i x_i + a_j C_j x_j,$$

and thus $\mathbf{x}'$ satisfies the constraint (4.3). However, the increment of the objective function is

$$z(\pi, \mathbf{x}') - z(\pi, \mathbf{x}) = w_i C_i x_i' + w_j C_j x_j' - w_i C_i x_i - w_j C_j x_j$$

$$= w_i C_i (x_i' - x_i) + w_j C_j (x_j' - x_j) = w_i C_i \frac{a_j C_j}{a_i C_i} \varepsilon - w_j C_j \varepsilon = a_j C_j \varepsilon \left(\frac{w_i}{a_i} - \frac{w_j}{a_j} \right) > 0.$$

Hence $z(\pi, \mathbf{x}') > z(\pi, \mathbf{x})$, contradicting the optimality of $\sigma = (\pi, \mathbf{x})$. □

By this lemma, we can find optimal solutions from among those satisfying (a)−(b). Now suppose that the jobs are indexed in the order that

$$\frac{w_1}{a_1} \le \frac{w_2}{a_2} \le \ldots \le \frac{w_n}{a_n}. \tag{4.5}$$

For a given schedule π, we can determine the completion time $C_j(\pi)$ of job j with respect to π $(1 \le j \le n)$ by

$$C_j(\pi) := \sum_{\pi^{-1}(i) \le \pi^{-1}(j)} p_i, \quad 1 \le j \le n. \tag{4.6}$$

Then the LP (4.2)–(4.4) for $\mathbf{x}$ can be written as

$$\max z(\pi, \mathbf{x}) = \textstyle\sum_{j=1}^n w_j C_j(\pi) x_j \tag{4.7}$$

$$\text{s.t.} \quad \textstyle\sum_{j=1}^n a_j C_j(\pi) x_j \le D \tag{4.8}$$

$$0 \le x_j \le 1, \quad 1 \le j \le n. \tag{4.9}$$

This LP is a Fractional Knapsack Problem which has the following simple property.

Lemma 4.3. *Suppose that the jobs are indexed by the order of* (4.5) *and*

$$k := \max \left\{ j \in \{1, 2, \ldots, n\} : \sum_{i=j}^n a_i C_i(\pi) \ge D \right\}.$$

Then an optimal solution of the fractional knapsack problem (4.7)–(4.9) *can be obtained by setting* $x_j = 1$ *for* $k+1 \le j \le n$, $x_k = \delta$, $x_i = 0$ *for* $1 \le i \le k-1$, *where*

$$\delta := \frac{D - \sum_{i=k+1}^n a_i C_i(\pi)}{a_k C_k(\pi)}.$$

Proof. See text books of LP or use the method of proving (b) in Lemma 4.2. □

To state more precisely, the LP (4.7)–(4.9) can be solved by the following procedure.

Procedure LP

(1) For a given schedule π, compute the completion time $C_j(\pi)$ of job j $(1 \le j \le n)$ by (4.6). Set $j := n$.

(2) **While** $D > 0$ **do**: **If** $a_j C_j(\pi) < D$, **then** set $x_j := 1$, $D := D - a_j C_j(\pi)$ and $j := j - 1$. **If** $a_j C_j(\pi) \ge D$, **then** set $x_j := \frac{D}{a_j C_j(\pi)} = \delta$ and $D := D - a_j C_j(\pi) x_j = 0$.

(3) Let $k := j$. Return the solution $\mathbf{x}$ by $x_j = 1$ for $k+1 \le j \le n$, $x_k = \delta$, $x_i = 0$ for $1 \le i \le k-1$.

In this procedure, we may call the job k the *bordering job*, which is the only possibility for $0 < x_k < 1$. In traditional theory of LP, the variable x_k here is the basic variable corresponding to the inequality (4.3) or (4.8).

On the other hand, suppose that the bordering job k is given. Then we can write

$$x_k = \frac{1}{a_k C_k(\pi)}(D - \sum_{j=k+1}^{n} a_j C_j(\pi) x_j) \tag{4.10}$$

from the constraint (4.8), where $x_j = 1$ for $k+1 \le j \le n$ and $0 < x_k \le 1$. In order to eliminate this constraint, we substitute (4.10) into the objective function (4.1) so that

$$\begin{aligned} T(\pi, \mathbf{x}) &= \sum_{i=1}^{k} w_i C_i(\pi) - w_k C_k(\pi) x_k \\ &= \sum_{i=1}^{k} w_i C_i(\pi) - \frac{w_k}{a_k}(D - \sum_{j=k+1}^{n} a_j C_j(\pi)) \\ &= \sum_{j=1}^{n} w'_j C_j(\pi) - \frac{w_k}{a_k} D, \end{aligned} \tag{4.11}$$

where

$$w'_j = \begin{cases} w_j, & \text{for } 1 \le j \le k \\ \dfrac{w_k a_j}{a_k}, & \text{for } k+1 \le j \le n. \end{cases} \tag{4.12}$$

Lemma 4.4. *Suppose that* $\mathbf{x}$ *is obtained by Procedure LP for a given schedule* π *and* k *is the bordering job in this procedure. Let* w'_j $(1 \le j \le n)$ *be the new weights defined by* (4.12). *Then the solution* $\sigma = (\pi, \mathbf{x})$ *is optimal if and only if* π *is in the WSPT order with respect to the new weights* w'_j, *that is,*

$$\frac{p_{\pi(1)}}{w'_{\pi(1)}} \le \frac{p_{\pi(2)}}{w'_{\pi(2)}} \le \dots \le \frac{p_{\pi(n)}}{w'_{\pi(n)}}.$$

Proof. Suppose that $\sigma = (\pi, \mathbf{x})$ is optimal. Then we can obtain a bordering job k by Procedure LP for π. By the above analysis, the objective function (4.1) is equivalent to (4.11). In the latter, $\frac{w_k}{a_k} D$ is a constant. So the problem is reduced to minimizing $\sum_{j=1}^{n} w'_j C_j(\pi)$, which is independent of the assignment $\mathbf{x}$. It is well known that a schedule π is optimal for the problem $1||\sum w_j C_j$ if and only if it is in the WSPT order. Therefore, π must be in the WSPT order with respect to the new weights w'_j.

Conversely, suppose that π is in the WSPT order with respect to the new weights w'_j. For the bordering job k, we have the basic variable representation (4.10), and so the objective function is equivalent to (4.11). Hence minimizing $T(\pi, \mathbf{x})$ is equivalent to minimizing $\sum_{j=1}^{n} w'_j C_j(\pi)$. It is known that the schedule π in the WSPT order with respect to w'_j is optimal for $\sum_{j=1}^{n} w'_j C_j(\pi)$. Therefore, $\sigma = (\pi, \mathbf{x})$ is optimal for $T(\pi, \mathbf{x})$. This completes the proof. □

As a result, in order to find an optimal solution, it suffices to look for a bordering job k such that the induced schedule π satisfies the condition of Lemma 4.4. We may call this k a *valid bordering job*. In the following algorithm, we use binary search to find this k in the range $1 \le k \le n$. During the search process, let l and u denote the lower and upper bounds.

Algorithm LP-WSPT

(0) Let $l := 1$ and $u := n$.
(1) If $l = u$, then $k = l$ is the valid bordering job. Go to Step (5).
(2) Let $k := \lfloor \frac{1}{2}(l+u) \rfloor$. Compute the new weights w'_j $(1 \le j \le n)$ by (4.12). Construct a schedule π in the WSPT order with respect to the weights w'_j.
(3) Run Procedure LP for the schedule π. Let k^* be the bordering job produced by this procedure. If $k^* = k$, then k is the valid bordering job. Go to Step (5)
(4) If $k < k^*$, then set $l := k+1$. If $k^* < k$, then set $u := k-1$. Go to Step (1).
(5) Return the optimal solution $\sigma = (\pi, \mathbf{x})$ by the valid bordering job.

Theorem 4.5. *Algorithm LP-WSPT correctly solves the problem* $1|\sum a_j d_j \le D|\sum w_j T_j$ *in* $O(n(\log n)^2)$ *time.*

Proof. By Lemma 4.4, if we can find a valid bordering job k, then $\sigma = (\pi, \mathbf{x})$ is an optimal solution, where π is constructed in Step (2) and $\mathbf{x} = (0, \dots, 0, x_k, 1, \dots, 1)$ is the assignment determined by the bordering job k. As the algorithm is a binary search, we need only show that the valid bordering job k is contained in the search range $[l, u]$. This is true at the beginning when $l = 1$ and $u = n$. In Step (3), if $k^* = k$, then k is a valid bordering job and we are done. If $k < k^*$, then

$$D - \sum_{j=k+1}^{n} a_j C_j(\pi) \le D - \sum_{j=k^*+1}^{n} a_j C_j(\pi) - a_{k^*} C_{k^*}(\pi) x_{k^*} = 0.$$

Thus the job k is impossible to be the bordering job of π (and of course not valid). For $k' < k$, let π' be the schedule constructed in Step (2) for k'. We show below that the job k' cannot be the bordering job of π' either. By the new weight formula (4.12), we have the new weights corresponding to k as

$$w_1, \dots, w_{k'}, w_{k'+1}, \dots, w_k, \frac{w_k a_{k+1}}{a_k}, \frac{w_k a_{k+2}}{a_k}, \dots, \frac{w_k a_n}{a_k},$$

and the new weights corresponding to k' as

$$w_1, \dots, w_{k'}, \frac{w_{k'} a_{k'+1}}{a_{k'}}, \dots, \frac{w_{k'} a_k}{a_{k'}}, \frac{w_{k'} a_{k+1}}{a_{k'}}, \frac{w_{k'} a_{k+2}}{a_{k'}}, \dots, \frac{w_{k'} a_n}{a_{k'}}.$$

Since $\frac{w_{k'}}{a_{k'}} \le \frac{w_k}{a_k}$, we have $\frac{w_{k'} a_j}{a_{k'}} \le \frac{w_k a_j}{a_k}$ for any j with $k+1 \le j \le n$. Especially, the order of these jobs $k+1, k+2, \dots, n$ in π is the same as that in π', because the ratios $\frac{p_j}{w'_j}$ are multiplied by a constant. Moreover, with regard to the orders between the jobs $k+1 \le j \le n$ and the others, we have the following observations:

- For any two jobs l and j with $1 \le l \le k', k+1 \le j \le n$, if job l is scheduled before job j in π, namely, $\frac{p_l}{w_l} \le \frac{a_k}{w_k} \cdot \frac{p_j}{a_j}$, then $\frac{p_l}{w_l} \le \frac{a_k}{w_k} \cdot \frac{p_j}{a_j} \le \frac{a_{k'}}{w_{k'}} \cdot \frac{p_j}{a_j}$. Hence job l is also scheduled before job j in π'.
- For any two jobs l and j with $k'+1 \le l \le k, k+1 \le j \le n$, if job l is scheduled before job j in π, namely, $\frac{p_l}{w_l} \le \frac{a_k}{w_k} \cdot \frac{p_j}{a_j}$, then by $\frac{w_k}{a_k} \ge \frac{w_l}{a_l}$, it holds that $\frac{p_j}{a_j} \ge \frac{p_l}{w_l} \cdot \frac{w_k}{a_k} \ge \frac{p_l}{a_l}$. Thus job l is also scheduled before job j in π'.

By the orders of π and π' mentioned above, we conclude that $C_j(\pi) \le C_j(\pi')$ for $k+1 \le j \le n$. Then

$$\sum_{j=k'+1}^{n} a_j C_j(\pi') > \sum_{j=k+1}^{n} a_j C_j(\pi') \ge \sum_{j=k+1}^{n} a_j C_j(\pi) \ge D.$$

Thus the job k' cannot be the bordering job of π'. Therefore, all jobs $k' \leq k$ should be eliminated from the search range by setting $l := k + 1$.

Likewise, if $k > k^*$, then

$$D - \sum_{j=k}^{n} a_j C_j(\pi) > D - \sum_{j=k^*+1}^{n} a_j C_j(\pi) - a_{k^*} C_{k^*}(\pi) x_{k^*} = 0.$$

Thus the job k cannot be the bordering job of π. Furthermore, it can be shown that the job $k' > k$ cannot be the bordering job of π' either (by the same argument as above). So we can eliminate all jobs $k' \geq k$ from the search range by setting $u := k - 1$.

For the running time of the algorithm, the computation for each k is called a *stage*. As we know, binary search has $\log n$ stages. In each stage, the main computation is included in Steps (2)$-$(3). In Step (2), computing the new weighs and constructing the schedule in WSPT order can be completed in $O(n \log n)$ time. In Step (3), Procedure LP can be carried out in $O(n)$ time. The remaining computations also take $O(n)$ time. Therefore, the overall complexity is $O(n(\log n)^2)$. □

5. Pareto optimization version

We proceed to consider the simultaneous optimization problem $1||(f_1, f_2)$ where $f_1 = \sum a_j d_j$, $f_2 = \sum w_j T_j$. The main theme is to apply the constraint version $1|\sum a_j d_j \leq D|\sum w_j T_j$. We suppose the jobs are indexed in the order that $\frac{w_1}{a_1} \leq \frac{w_2}{a_2} \leq \ldots \leq \frac{w_n}{a_n}$. Then the constraint version is

$$\begin{aligned} &\min \textstyle\sum_{j=1}^{n} w_j(1 - x_j) C_j(\pi) \\ &\text{s.t.} \quad \textstyle\sum_{j=1}^{n} a_j C_j(\pi) x_j \leq D \\ &\qquad 0 \leq x_j \leq 1, \quad 1 \leq j \leq n, \end{aligned}$$

where π is an unknown schedule, $d_j = C_j(\pi) x_j$, and $D \geq 0$ is a parameter. We denote this problem by CV(D).

By Algorithm LP-WSPT in the previous section, an optimal solution of problem CV(D) is $\sigma = (\pi, \mathbf{d})$ where π is a schedule in the WSPT order with the new weights defined in (4.12) and $\mathbf{d}$ is in the form

$$\mathbf{d} = (0, \ldots, 0, C_k(\pi)\delta, C_{k+1}(\pi), \ldots, C_n(\pi)),$$

where k is the bordering job ($1 \leq k \leq n$) and $D = a_k C_k(\pi)\delta + \sum_{j=k+1}^{n} a_j C_j(\pi)$.

Lemma 5.1. *For any given D, the optimal solution $\sigma = (\pi, \mathbf{d})$ of problem CV(D) produces a Pareto optimal solution for the bicriteria scheduling problem $1||(f_1(\sigma), f_2(\sigma))$.*

Proof. If $\sigma = (\pi, \mathbf{d})$ is not a Pareto optimal solution, then there exists another solution $\sigma' = (\pi', \mathbf{d}')$ such that $\sum_{1 \leq j \leq n} a_j d'_j \leq \sum_{1 \leq j \leq n} a_j d_j$ and $\sum_{1 \leq j \leq n} w_j T'_j \leq \sum_{1 \leq j \leq n} w_j T_j$ where at least one of the inequalities is strict. We distinguish two cases as follows.

(i) $\sum_{1 \leq j \leq n} a_j d'_j \leq \sum_{1 \leq j \leq n} a_j d_j \leq D$ and $\sum_{1 \leq j \leq n} w_j T'_j < \sum_{1 \leq j \leq n} w_j T_j$. This contradicts that $\sigma = (\pi, \mathbf{d})$ is an optimal solution of problem CV(D).

(ii) $\sum_{1 \leq j \leq n} a_j d'_j < \sum_{1 \leq j \leq n} a_j d_j \leq D$ and $\sum_{1 \leq j \leq n} w_j T'_j = \sum_{1 \leq j \leq n} w_j T_j$. Let $D^* = \sum_{1 \leq j \leq n} a_j d'_j$ and let $\sigma^* = (\pi^*, \mathbf{d}^*)$ be the optimal solution of problem CV(D^*). Then $T(\pi^*, \mathbf{d}^*) \leq T(\pi', \mathbf{d}')$, namely $\sum_{1 \leq j \leq n} w_j T^*_j \leq \sum_{1 \leq j \leq n} w_j T'_j$. Besides, in performing Algorithm LP-WSPT for D^*, let k^* be the bordering job and x^*_j the corresponding variables. On the other hand, let $\tilde{k}$ be the the bordering job in Procedure LP for D and the given schedule π^*. Then we obtain a feasible solution $(\pi^*, \tilde{\mathbf{d}})$ by this bordering job $\tilde{k}$. Since $D^* < D$, it follows that $k^* \geq \tilde{k}$ and $x^*_{k^*} < x_{\tilde{k}}$ if $k^* = \tilde{k}$. Hence

$$T(\pi^*, \tilde{\mathbf{d}}) = \sum_{i=1}^{\tilde{k}} w_i C_i(\pi^*) - w_{\tilde{k}} C_{\tilde{k}}(\pi^*) x_{\tilde{k}} < \sum_{i=1}^{k^*} w_i C_i(\pi^*) - w_{k^*} C_{k^*}(\pi^*) x^*_{k^*} = T(\pi^*, \mathbf{d}^*).$$

Therefore, $\sum_{1\leq j\leq n} w_j \tilde{T}_j < \sum_{1\leq j\leq n} w_j T^*_j \leq \sum_{1\leq j\leq n} w_j T'_j = \sum_{1\leq j\leq n} w_j T_j$, contradicting the optimality of $\sigma = (\pi, \mathbf{d})$. This completes the proof. □

Lemma 5.2. *Any Pareto optimal point of bicriteria scheduling problem* $1||(f_1(\sigma), f_2(\sigma))$ *can be generated by an optimal solution of problem CV(D).*

Proof. Suppose that $\sigma^0 = (\pi^0, \mathbf{d}^0)$ is a Pareto optimal solution of $1||(f_1(\sigma), f_2(\sigma))$. Let $D^0 = \sum_{1\leq j\leq n} a_j d^0_j$. Then σ^0 is an optimal solution of problem CV(D^0) by the definition of Pareto optimality of σ^0. Let $\sigma = (\pi, \mathbf{d})$ be the optimal solution of problem CV(D^0) produced by Algorithm LP-WSPT. Then $T(\pi^0, \mathbf{d}^0) = T(\pi, \mathbf{d})$ and $\sum_{1\leq j\leq n} a_j d^0_j = D^0 = \sum_{1\leq j\leq n} a_j d_j$ by the consequence of Algorithm LP-WSPT. This gives $f_1(\sigma^0) = f_1(\sigma)$ and $f_2(\sigma^0) = f_2(\sigma)$. Thus the Pareto optimal point $(f_1(\sigma^0), f_2(\sigma^0)) = (f_1(\sigma), f_2(\sigma))$ can be generated by the optimal solution of Algorithm LP-WSPT with respect to CV(D^0). □

By these lemmas, we conclude that all Pareto optimal solutions can be obtained by solving the problems CV(D) for different D. When $D = 0$, we have the first Pareto optimal point $(0, \sum_{i=1}^n w_{\pi(i)} C_{\pi(i)})$, where π is a schedule in the WSPT order. As D increases, there are infinite Pareto optimal points which constitute a trade-off curve of piecewise linear function. We will see that this piecewise linear function has at most n linear segments, each of which corresponds to a bordering job k.

For each bordering job k with $1 \leq k \leq n$, let $\pi^{(k)}$ be the schedule produced by Algorithm LP-WSPT, that is, in the WSPT order with the new weights defined in (4.12). Meanwhile, the due date assignments for different δ $(0 \leq \delta \leq 1)$ are

$$\mathbf{d}^{(k)}(\delta) = (0, \ldots, 0, C_k(\pi^{(k)})\delta, C_{k+1}(\pi^{(k)}), \ldots, C_n(\pi^{(k)})). \tag{5.1}$$

Then the criterion for due date cost is

$$f_1(\pi^{(k)}, \mathbf{d}^{(k)}(\delta)) = \sum_{j=1}^{n} a_j d_j = a_k C_k(\pi^{(k)})\delta + \sum_{j=k+1}^{n} a_j C_j(\pi^{(k)}), \tag{5.2}$$

which is a linear function on δ with $0 \leq \delta \leq 1$. Accordingly, the criterion for the total weighted tardiness is

$$f_2(\pi^{(k)}, \mathbf{d}^{(k)}(\delta)) = T(\pi^{(k)}, \mathbf{d}^{(k)}(\delta)) = \sum_{i=1}^{k} w_i C_i(\pi^{(k)}) - w_k C_k(\pi^{(k)})\delta, \tag{5.3}$$

which is a linear function on δ with $0 \leq \delta \leq 1$. This amounts to solving the constraint problem CV(D) with $D = a_k C_k(\pi^{(k)})\delta + \sum_{j=k+1}^{n} a_j C_j(\pi^{(k)})$.

To summarize, we have an algorithm for constructing the trade-off curve of problem $1||(\sum a_j d_j, \sum w_j T_j)$ as follows.

Algorithm PRT-OPT

(1) Re-index the jobs such that

$$\frac{w_1}{a_1} \leq \frac{w_2}{a_2} \leq \ldots \leq \frac{w_n}{a_n}\cdot$$

Let $k := n$.

(2) For the bordering job k, construct the schedule $\pi^{(k)}$ in the WSPT order with the new weights defined in (4.12), and construct the assignment $\mathbf{d}^{(k)}(\delta)$ by (5.1).
(3) Compute $f_1(\pi^{(k)}, \mathbf{d}^{(k)}(\delta))$ by (5.2) and compute $f_2(\pi^{(k)}, \mathbf{d}^{(k)}(\delta))$ by (5.3).
(4) Return the linear segments of Stage k:

$$(f_1(\pi^{(k)}, \mathbf{d}^{(k)}(\delta)), f_2(\pi^{(k)}, \mathbf{d}^{(k)}(\delta))), \quad 0 \leq \delta \leq 1.$$

If $k = 1$, then stop; otherwise set $k := k - 1$ and go to Step (2).

Theorem 5.3. *Algorithm PRT-OPT correctly solves the problem* $1||(\sum a_j d_j, \sum w_j T_j)$ *by determining the trade-off curve in* $O(n^2 \log n)$ *time.*

Proof. By Lemmas 5.1 and 5.2, the trade-off curve containing all Pareto optimal points can be constructed by the solutions of the problems CV(D) for all possible $D \geq 0$. For this purpose, Algorithm PRT-OPT produces all these solutions $\sigma^{(k)} = (\pi^{(k)}, \mathbf{d}^{(k)}(\delta))$ $(0 \leq \delta \leq 1)$ for $1 \leq k \leq n$.

We next analysis the running time of the algorithm. In Step (1), we define the initial schedule in $O(n \log n)$ time. The algorithm consists of n stages, each of which (Stage k) includes Steps (2)−(4). In Step (2), we construct the schedule $\pi^{(k)}$ in $O(n \log n)$ time and construct the assignment $\mathbf{d}^{(k)}(\delta)$ in $O(n)$ time. In Steps (3)−(4), computing $f_1(\pi^{(k)}, \mathbf{d}^{(k)}(\delta))$ and $f_2(\pi^{(k)}, \mathbf{d}^{(k)}(\delta))$ can be completed in $O(n)$ time. Therefore, the overall complexity of n stages is $O(n^2 \log n)$. □

6. Concluding remarks

Multicriteria and Multiagent scheduling [1,17] is an active area in modern scheduling theory. It is meaningful to combine the due date assignment with this area. The bicriteria scheduling of due date assignment has two objective functions, one is the due date cost, another is a due date involving criterion in scheduling system. For the second criterion, $\sum w_j U_j$ (the weighted number of tardy jobs), $T_{\max}$ (the maximum tardiness) and $\sum w_j T_j$ (the total weighted tardiness) have been investigated. More second criteria, such as the earliness and tardiness, should be further studied.

In the foregoing discussion, we only consider that the due date cost is a linear function $\sum_{j=1}^{n} a_j d_j$. Similar to [14], the due date cost may have more complicated form, such as $\sum_{j=1}^{n} f_j(d_j)$. We believe that the problems would be changed to be NP-hard for this generalized cost. Also, it is worthwhile to study the case where some constraints are imposed to the due dates.

Acknowledgements. The authors would like to thank the referees for their helpful comments on improving the presentation of the paper.

References

[1] A. Agnetis, J.-C. Billaut, S. Gawiejnowicz, D. Pacciarelli and A. Soukhal, Multiagent Scheduling: Models and Algorithms. Springer-Verlag, Berlin (2014).
[2] J. Błazewicz, K. Ecker, G. Schmidt and J. Węglarz, Scheduling in Computer and Manufacturing Systems. Springer-Verlag, Berlin (1993).
[3] P. Brucker, Scheduling Algorithms 3rd Edition. Springer-Verlag, Berlin (2001).
[4] T.C.E. Cheng and M.C. Gupta, Survey of scheduling research involving due date determination decisions. *Eur. J. Operat. Res.* **38** (1989) 156–166.
[5] V. Gordan and W. Kubiak, Single machine scheduling with release and due date assignment to minimize the weighted number of late jobs. *Infor. Proc. Lett.* **68** (1998) 153–159.
[6] V. Gordan, J.M. Proth and C. Chu, A survey of the state-of-the-art of common due date assignment and scheduling research. *Eur. J. Oper. Res.* **139** (2002) 1–25.
[7] V. Gordan and V. Strusevich, Single machine scheduling and due date assignment with positionally dependent processing times. *Eur. J. Oper. Res.* **198** (2009) 57–62.
[8] V. Gordan, V. Strusevich and A. Polgui, Scheduling with due date assignment under special conditions on job processing. *J. Scheduling* **15** (2012) 447–456.

[9] H. Hoogeveen, Multicriteria scheduling. *Eur. J. Oper. Res.* **167** (2005) 592–623.
[10] C. Koulamas, A faster algorithm for a due date assignment problem with tardy jobs. *Oper. Res. Lett.* **38** (2010) 127–128.
[11] H. Lin and C. He, On the bicriteria scheduling of due date assignment and weighted number of tardy jobs. Chinese. *J. Eng. Math.* **34** (2017) 73–84.
[12] D. Shabtay and G. Steiner, Two due date assignment problems in scheduling a single machine. *Oper. Res. Lett.* **34** (2006) 683–691.
[13] D. Shabtay and G. Steiner, The single-machine earliness-tardiness scheduling problem with due date assignment and resource-dependent processing times. *Ann. Oper. Res.* **159** (2008) 25–40.
[14] D. Shabtay, G. Steiner and L. Yedidsion, Bicriteria problems to minimize maximum tardiness and due date assignment cost in various scheduling enviroments. *Discrete Appl. Math.* **158** (2010) 1090–1103.
[15] D. Shabtay and G. Steiner, A bicriteria approach to minimize the total weighted number of tardy jobs with convex controllable processing times and assignable due dates. *J. Scheduling* **14** (2011) 455–469.
[16] D. Shabtay, Optimal restricted due date assignment in scheduling. *Eur. J. Oper. Res.* **252** (2016) 79–89.
[17] V. T'kindt and J.-C. Billaut, Multicriteria Scheduling: Theory, Models and Algorithms, 2nd Edition. Springer-Verlag, Berlin (2006).