Two-agent preemptive Pareto-scheduling to minimize the number of tardy jobs and total late work

Ruyan He[1], **Jinjiang Yuan**[1*], **C.T. Ng**[2], **T.C.E. Cheng**[2]

[1]School of Mathematics and Statistics, Zhengzhou University,

Zhengzhou, Henan 450001, People's Republic of China

[2]Logistics Research Centre, Department of Logistics and Maritime Studies,

The Hong Kong Polytechnic University, Hong Kong SAR, People's Republic of China

---

Abstract We consider the single-machine preemptive Pareto-scheduling problem with two competing agents A and B, where agent A wants to minimize the number of its jobs (the A-jobs) that is tardy, while agent B wants to minimize the total late work of its jobs (the B-jobs). We provide an $O(nn_A \log n_A + n_B \log n_B)$-time algorithm that generates all the Pareto-optimal points, where n_A is the number of the A-jobs, n_B is the number of the B-jobs, and $n = n_A + n_B$.

Keywords: scheduling; two agents; Pareto-scheduling; number of tardy jobs; total late work.

1 Introduction

Background: Scheduling has remained an active domain of research in the field of combinatorial optimization and an abundance of theoretical results have been developed over the years (see, e.g., Brucker (2001), Chen et al. (1998), and Lawler (1983)). Motivated by a wide array of practical applications, researchers have developed a great variety

*Corresponding author. Email address: yuanjj@zzu.edu.cn

of scheduling models and made remarkable progress in addressing their computational complexity issues, and devising solution and approximation algorithms for them. One main branch of scheduling research concerns multi-agent scheduling, in which two-agent scheduling is at its core. In the literature, two-agent scheduling that considers the late work criterion is an emerging topic that has not been extensively studied. Considering this topic, we study the two-agent preemptive Pareto-scheduling problem, in which one agent wants to minimize the number of its tardy jobs, while the other agent wants to minimize the total late work of its jobs.

Problem Formulation: We introduce the single-machine preemptive scheduling problem with two competing agents as follows: There are two agents A and B that compete to perform their respective jobs on a common machine. Let $\mathcal{J}^A = \{J_1^A, J_2^A, \ldots, J_{n_A}^A\}$ and $\mathcal{J}^B = \{J_1^B, J_2^B, \ldots, J_{n_B}^B\}$ denote the job sets of agent A and agent B, respectively, under the competing restriction that $\mathcal{J}^A \cap \mathcal{J}^B = \emptyset$. For each agent $X \in \{A, B\}$, the jobs of $\mathcal{J}^X$ are called the X-jobs. Each job J_j^X has a processing time $p_j^X > 0$ and a due date $d_j^X \geq 0$, both of which are integer-valued. Let $P_X = \sum_{j=1}^{n_X} p_j^X$ for $X \in \{A, B\}$, $n = n_A + n_B$, $\mathcal{J} = \mathcal{J}^A \cup \mathcal{J}^B$, and $P = P_A + P_B$. All the jobs are available at time zero. We assume that the maximum due date of all the jobs is at most P and the schedules are preemptive. A feasible schedule requires that any pair of jobs cannot be processed in the same time slot.

Given a feasible schedule σ of the n independent jobs of $\mathcal{J}$, we use $C_j^X(\sigma)$ to denote the *completion time* of a job J_j^X, $X \in \{A, B\}$. The *late work* of job J_j^X under σ, denoted by $Y_j^X(\sigma)$, is the amount of processing of J_j^X after its due date d_j^X in σ. If $Y_j^X(\sigma) = 0$ or, equivalently, $C_j^X(\sigma) \leq d_j^X$, then J_j^X is called *early* under σ. If $0 < Y_j^X(\sigma) < p_j^X$, then J_j^X is called *partially early* under σ. If $Y_j^X(\sigma) = p_j^X$, then J_j^X is called *late* under σ. J_j^X is called *non-late* under σ if J_j^X is either early or partially early, i.e., not late, under σ. J_j^X is called *tardy* under σ if $C_j^X(\sigma) > d_j^X$, i.e., J_j^X is either partially early or late under σ. We define $U_j^X(\sigma) = 0$ if J_j^X is early under σ and define $U_j^X(\sigma) = 1$ if J_j^X is tardy under σ. When no ambiguity may occur, we abbreviate $C_j^X(\sigma)$, $Y_j^B(\sigma)$, and $U_j^A(\sigma)$ as C_j^X, Y_j^B, and U_j^A, respectively. In addition, we use the the following notation throughout the paper.

- $\sum U_j^A = \sum_{j=1}^{n_A} U_j^A(\sigma)$ is the *number of tardy A-jobs* under schedule σ.
- $\sum Y_j^B = \sum_{j=1}^{n_B} Y_j^B(\sigma)$ is the *total late work* of the B-jobs under schedule σ.

Specifically, we consider in this paper the single-machine preemptive Pareto-scheduling problem with two competing agents A and B to minimize the number of tardy A-jobs and total late work of the B-jobs. The goal is to find all the Pareto-optimal points and, for each Pareto-optimal point, the corresponding Pareto-optimal schedule. T'kindt and

Billaut (2006) gave the formal definitions of Pareto-optimal points and Pareto-optimal schedules. Following the notation in T'kindt and Billaut (2006), we denote the scheduling problem under study as

$$1|\text{pmtn}|^{\#}(\sum U_j^A, \sum Y_j^B). \tag{1}$$

Throughout the paper, we sort the A-jobs such that

$$d_1^A \leq d_2^A \leq \cdots \leq d_{n_A}^A \leq P \tag{2}$$

with ties being broken by the longest processing time (LPT) first rule, i.e.,

$$i < j \text{ if } d_i^A = d_j^A \text{ and } p_i^A > p_j^A. \tag{3}$$

The assumption of preemptive scheduling and the criterion $\sum Y_j^B$ of agent B enable us to make a simple assumption for the B-jobs. If there are two B-jobs J_j^B and $J_{j'}^B$ such that $d_j^B = d_{j'}^B$, then we can merge J_j^B and $J_{j'}^B$ into a new job $J_{j \cdot j'}^B$ such that $p_{j \cdot j'}^B = p_j^B + p_{j'}^B$ and $d_{j \cdot j'}^B = d_j^B$. This clearly does not affect the results of our analysis of the problem. So we assume that the B-jobs have distinct due dates. Throughout the paper, we sort the B-jobs such that

$$d_1^B < d_2^B < \cdots < d_{n_B}^B \leq P. \tag{4}$$

Literature Review: There is a large body of literature on scheduling involving two competing agents, scheduling considering the tardy-job criterion, and scheduling considering the late work criterion. For our purpose, we only review the most related results.

Agnetis et al. (2004), and Baker and Smith (2003) pioneered two-agent scheduling research. Agnetis et al. (2004) considered various constrained scheduling problems and Pareto-scheduling problems involving two competing agents. The objective functions they sought to minimize include the maximum scheduling cost, total (weighted) completion time, and number of tardy jobs. Baker and Smith (2003) considered the global objective to minimize a positive combination of the criteria of two competing agents. The objective functions they sought to minimize include three basic scheduling criteria, namely the makespan, maximum lateness, and total weighted completion time. Yuan et al. (2005a) further studied some of the problems that Baker and Smith (2003) considered. For detailed results on two-agent scheduling, we refer the reader to Agnetis et al. (2014), Cheng et al. (2006, 2008), Lee et al. (2010), Leung et al. (2010), Liu et al. (2019), Ng et al. (2006), Oron et al. (2015), and Yuan (2016, 2018).

Blazewicz and Finke (1987) first studied scheduling considering the late work criterion. They showed that the preemptive scheduling problem with release dates on identical parallel machines can be solved by linear programming, so the problem is polynomially solvable. Potts and Van Wassenhove (1991a) considered single-machine scheduling to minimize the total late work. They showed that the problem is NP-hard and is solvable in pseudo-polynomial time. Potts and Van Wassenhove (1991b) further proposed a branch-bound algorithm and presented two fully polynomial-time approximation schemes, which run in $O(\frac{n^2}{\epsilon})$ and $O(\frac{n^3}{\epsilon})$ times, respectively, for the problem. Hariri et al. (1995) considered single-machine preemptive scheduling to minimize the total weighted late work and presented an $O(n \log n)$-time algorithm for the problem. Zhang and Wang (2017) studied the two-agent scheduling problem where the objective is to minimize the total weighted late work of agent A's jobs, while keeping the maximum cost of agent B within a given bound U. They analyzed the computational complexity of three related problems, and presented polynomial-time or pseudo-polynomial-time algorithms to solve them. Zhang and Yuan (2019) further studied the problems that Zhang and Wang (2017) considered. Chen et al. (2019) studied single-machine scheduling to minimize the total weighted late work with job deadlines. They showed that the problem is unary NP-hard even if all the jobs have a unit weight, the problem is binary NP-hard and admits a pseudo-polynomial-time algorithm and a fully polynomial-time approximation scheme if all the jobs have a common due date, and some special cases of the problem are polynomially solvable. Recently, He and Yuan (2019) studied two-agent preemptive Pareto-scheduling where one agent aims to minimize the total late work of its jobs, while the other agent aims to minimize the total late work, maximum lateness, or total completion time of its jobs. They presented three corresponding polynomial-time solution algorithms that find the trade-off curves. For more results on scheduling research to minimize the total late work, we refer the reader to the survey paper of Sterna (2011).

For the classical single-machine scheduling problem $1||\sum U_j$, Moore (1968) provided an $O(n \log n)$-time solution algorithm, which is known as the Moore's algorithm. Note that problem $1||\sum U_j$ is equivalent to its preemptive version $1|\text{pmtn}|\sum U_j$, so the latter is also solvable by Moore's algorithm in $O(n \log n)$ time. Recently, Zhao and Yuan (2019) presented another algorithm, named BSRPT, to solve problem $1|\text{pmtn}|\sum U_j$ in $O(n \log n)$ time. Wan et al. (2016) presented a polynomial-time algorithm to solve the two-agent Pareto-scheduling problem $1||^{\#}(\sum U_j^A, f_{\max}^B)$. As an intermediate result, they presented an $O(n \log n)$-time algorithm to solve the problem of scheduling n jobs to minimize the total processing time of the early jobs under the restriction that exactly k jobs are early, where $1 \leq k \leq n$. We deploy the algorithm in Wan et al. (2016) and a generalization of the algorithm BSRPT in Zhao and Yuan (2019) as subroutines in our solution algorithm.

Our Contributions: We study in this paper the single-machine two-agent preemtive Pareto-scheduling problem introduced in (1), i.e., $1|\mathrm{pmtn}|^{\#}(\sum U_j^A, \sum Y_j^B)$, and present an $O(nn_A \log n_A + n_B \log n_B)$-time algorithm to generate all the Pareto-optimal points. We make repeated use of the techniques in Wan et al. (2016), He and Yuan (2019), and Zhao and Yuan (2019) to address the problem.

We organize the rest of the paper as follows: In Section 2 we present some important lemmas and a critical algorithm for the constrained version of the problem under study. In Section 3 we introduce two subroutines for use in our algorithm. In Section 4 we present a method to determine all the Pareto-optimal points. In Section 5 we provide a polynomial-time algorithm to generate all the Pareto-optimal points for our problem.

2 Preliminaries

In this section we present some preliminary results necessary for the analyses in Sections 4 and 5.

2.1 Pareto optimality

For the two-agent preemtive Pareto-scheduling problem $1|\mathrm{pmtn}|^{\#}(\sum U_j^A, \sum Y_j^B)$ on the instance $\mathcal{J} = \mathcal{J}^A \cup \mathcal{J}^B$, we use $\Omega(\mathcal{J}^A, \mathcal{J}^B)$ to denote the set of all the Pareto-optimal points. Recall from (4) that $d_1^B < d_2^B < \cdots < d_{n_B}^B$. Throughout the paper, we define

$$\sigma_0^B = (J_1^B, J_2^B, \ldots, J_{n_B}^B). \tag{5}$$

From (4) and (5), the B-jobs are scheduled in the EDD order consecutively in σ_0^B. Thus, σ_0^B is an optimal schedule for the scheduling problem $1||T_{\max}^B$. From Potts and Van Wassenhove (1991a), the optimal value of problem $1|\mathrm{pmtn}|\sum Y_j^B$ on instance $\mathcal{J}^B$ is given by $T_{\max}(\sigma_0^B)$. For convenience, we set $Y^{(0)} = T_{\max}(\sigma_0^B)$. Then we have the following lemma.

Lemma 2.1. *For each point $(u, y) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$, $Y^{(0)} \leq y \leq P_B$.*

To study problem $1|\mathrm{pmtn}|^{\#}(\sum U_j^A, \sum Y_j^B)$, technically we need the constrained scheduling problem $1|\mathrm{pmtn}|\sum U_j^A : \sum Y_j^B \leq y$ on instance $\mathcal{J} = \mathcal{J}^A \cup \mathcal{J}^B$, which seeks to find a feasible schedule σ such that $\sum U_j^A(\sigma)$ is minimized, subject to the constraint that

$\sum Y_j^B(\sigma) \le y$. The feasibility of the problem requires that $y \ge Y^{(0)}$. For each $y \ge Y^{(0)}$, we use $U(y)$ to denote the optimal value of this constrained scheduling problem. The following lemma, which is implied in T'kindt and Billaut (2006), is useful for our research.

Lemma 2.2. *For each point $(u, y) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$, $u = U(y)$ and every optimal schedule for problem $1|pmtn|\sum U_j^A : \sum Y_j^B \le y$ is also a Pareto-optimal schedule corresponding to (u, y). Moreover, for each value $y \ge Y^{(0)}$, $(U(y), y') \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$, where y' is the minimum value in $[Y^{(0)}, y]$ such that $U(y') = U(y)$. This further implies that $(U(Y^{(0)}), Y^{(0)}) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$.*

Based on Lemma 2.2, we first consider the constrained problem $1|\text{pmtn}|\sum U_j^A$: $\sum Y_j^B \le y$.

2.2 Scheduling the B-jobs as forbidden intervals

Suppose that we have n jobs $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ to be scheduled preemptively on a single machine, and there is a set of m forbidden intervals $\mathcal{I} = \{h_k = [\tau_1^{(k)}, \tau_2^{(k)}] : k = 1, 2, \ldots, m\}$ in which no job can be scheduled, where $\tau_1^{(1)} < \tau_2^{(1)} < \tau_1^{(2)} < \tau_2^{(2)} < \cdots < \tau_1^{(m)} < \tau_2^{(m)}$. Throughout the paper, we use $|h_k|$ to denote the length of the interval h_k, i.e., $|h_k| = \tau_2^{(k)} - \tau_1^{(k)}$ for $k = 1, 2, \ldots, m$. When there is no risk of confusion, we also write $\mathcal{I} = \cup_{k=1}^m [\tau_1^{(k)}, \tau_2^{(k)}]$. Then the n jobs must be scheduled in the time space $[0, +\infty) \setminus \mathcal{I}$. We denote the single-machine preemptive scheduling problem with forbidden intervals to minimize the criterion f as $(1, \mathcal{I})|\text{pmtn}|f$.

Now let $\mathcal{J} = \mathcal{J}^A \cup \mathcal{J}^B$ and let f^A be an arbitrary regular scheduling criterion of agent A. Given a threshold value $y \in [Y^{(0)}, P_B]$, we consider the more general constrained problem $1|\text{pmtn}|f^A : \sum Y_j^B \le y$ on instance $\mathcal{J}$, which seeks to find a feasible schedule σ to minimize $f^A(\sigma)$, subject to the constraint that $\sum Y_j^B(\sigma) \le y$. For this constrained problem, He and Yuan (2019) showed that the B-jobs can be first scheduled before the A-jobs are scheduled.

To schedule the B-jobs, we use $j(y) \in \{1, 2, \ldots, n_B\}$ to denote the unique index such that $\sum_{j=1}^{j(y)-1} p_j^B < y \le \sum_{j=1}^{j(y)} p_j^B$; in case $y = 0$, which requires $Y^{(0)} = 0$, we just define $j(y) = 1$. We call $J_{j(y)}^B$ the *critical B-job* with respect to y. We decompose the critical B-job $J_{j(y)}^B$ into two parts $J_{j(y)}^{BE}$ and $J_{j(y)}^{BY}$ such that

$$p_{j(y)}^{BY} = y - \sum_{j=1}^{j(y)-1} p_j^B \text{ and } p_{j(y)}^{BE} = p_{j(y)}^B - p_{j(y)}^{BY} = \sum_{j=1}^{j(y)} p_j^B - y.$$

We call $J^{BE}_{j(y)}$ and $J^{BY}_{j(y)}$ the *early part* and the *late part* of $J^B_{j(y)}$, respectively, corresponding to y. Set $\mathcal{J}^{BE}(y) = \{J^{BE}_{j(y)}, J^B_{j(y)+1}, J^B_{j(y)+2}, \ldots, J^B_{n_B}\}$ and $\mathcal{J}^{BY}(y) = \{J^B_1, J^B_2, \ldots, J^B_{j(y)-1}, J^{BY}_{j(y)}\}$. Then the parts of $\mathcal{J}^{BE}(y)$ have the total processing time $P_B - y$ and the parts of $\mathcal{J}^{BY}(y)$ have the total processing time y.

He and Yuan (2019) provided a procedure, called Procedure(y), to schedule the B-jobs preemptively such that all the parts of $\mathcal{J}^{BE}(y)$ will be early and all the parts of $\mathcal{J}^{BY}(y)$ will be late. Recall that $y \in [Y^{(0)}, P_B]$.

Procedure(y): *Determine $j(y)$, $J^B_{j(y)}$, $J^{BE}_{j(y)}$, $J^{BY}_{j(y)}$, $\mathcal{J}^{BE}(y)$, and $\mathcal{J}^{BY}(y)$. Generate a schedule $\sigma^{B(y)}$ for the B-jobs $\mathcal{J}^B = \mathcal{J}^{BE}(y) \cup \mathcal{J}^{BY}(y)$ in the following way:*

(i) From time $P^* > d^B_{n_B}$, schedule the parts of $\mathcal{J}^{BY}(y)$ consecutively in the order $J^B_1, J^B_2, \ldots, J^B_{j(y)-1}, J^{BY}_{j(y)}$.

(ii) Schedule the parts of $\mathcal{J}^{BE}(y)$ using the algorithm of Hariri et al. (1995) for solving problem $1|pmtn|\sum Y_j$ on instance $\mathcal{J}^{BE}(y)$, which can be stated as follows:

Beginning with time $d^B_{n_B}$, schedule the parts of $\mathcal{J}^{BE}(y)$ backwards in the order

$$J^B_{n_B}, J^B_{n_B-1}, \ldots, J^B_{j(y)+1}, J^{BE}_{j(y)}$$

such that each part of $\mathcal{J}^{BE}(y)$ is scheduled as late as possible, subject to its due date.

By setting $P^* = P+1$, He and Yuan (2019) showed that, for every problem $1|\text{pmtn}|f^A : \sum Y^B_j \leq y$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$, where f^A is a regular criterion for the A-jobs, there exists an optimal schedule in which the B-jobs are scheduled by Procedure(y).

For the problems studied in this paper, any part of a job scheduled after time $\max\{d^A_{n_A}, d^B_{n_B}\}$ must be late. Then we define $P^* = 1 + \max\{d^A_{n_A}, d^B_{n_B}\}$ and use this notation throughout the paper.

Let $\sigma^{B(y)}$ be the schedule for the B-jobs generated by Procedure(y). Then we have the following lemma for our problem $1|\text{pmtn}|\sum U^A_j : \sum Y^B_j \leq y$.

Lemma 2.3. *Consider problem $1|pmtn|\sum U^A_j : \sum Y^B_j \leq y$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$. There exists an optimal schedule for the problem in which the B-jobs are scheduled in the same manner as that in $\sigma^{B(y)}$.*

For each $y \in [Y^{(0)}, P_B]$, we use $\mathcal{I}^{B(y)}$ to denote the set of time intervals occupied by the B-jobs in schedule $\sigma^{B(y)}$. For scheduling the A-jobs, we regard each interval of $\mathcal{I}^{B(y)}$ as a forbidden interval, which cannot be occupied by any A-job. Note that $(1, \mathcal{I}^{B(y)})|\text{pmtn}|\sum U^A_j$ is the single-machine preemptive scheduling problem to minimize $\sum U^A_j$ with the set of forbidden intervals $\mathcal{I}^{B(y)}$. From Lemma 2.3, the problem has the optimal value $U(y)$.

For a schedule π on $\mathcal{J} = \mathcal{J}^A \cup \mathcal{J}^B$, we use π^A to denote the subschedule of π for the A-jobs and use π^B to denote the subschedule of π for the B-jobs. A schedule π of $\mathcal{J}^A \cup \mathcal{J}^B$ is called *y-optimal* if $\pi^B = \sigma^{B(y)}$ and π^A is optimal for problem $(1, \mathcal{I}^{B(y)})|\text{pmtn}|\sum U_j^A$. From Lemmas 2.2 and 2.3, we have the following result.

Lemma 2.4. *Consider problem $1|pmtn|\sum U_j^A : \sum Y_j^B \le y$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$ and let π be a y-optimal schedule. Then π is optimal for the problem. Moreover, if (u, y) is a Pareto-optimal point, then π is a Pareto-optimal schedule corresponding to (u, y).*

2.3 The structure of the forbidden intervals

We first consider the forbidden intervals set $\mathcal{I}^{B(Y^{(0)})}$ that consists of the time intervals occupied by the B-jobs in schedule $\sigma^{B(Y^{(0)})}$. Following He and Yuan (2019), we assume that

$$\mathcal{I}^{B(Y^{(0)})} = \{h_1, h_2, \ldots, h_m\},$$

where $h_i = [\tau_1^{(i)}, \tau_2^{(i)}]$ is the i-th interval, $i = 1, 2, \ldots, m$, such that

$$0 \le \tau_1^{(1)} < \tau_2^{(1)} < \tau_1^{(2)} < \tau_2^{(2)} < \cdots < \tau_1^{(m)} < \tau_2^{(m)}. \tag{6}$$

From the implementation of Procedure($Y^{(0)}$), we have

$$\tau_1^{(m)} = P^* \text{ and } \tau_2^{(m)} = P^* + Y^{(0)}. \tag{7}$$

For each $y \in [Y^{(0)}, P_B]$, we define $i(y)$ as the maximum index in $\{1, 2, \ldots, m-1\}$ such that $y - Y^{(0)} \ge \sum_{i=1}^{i(y)-1}(\tau_2^{(i)} - \tau_1^{(i)})$ and let $\tau(y) \in [\tau_1^{(i(y))}, \tau_2^{(i(y))})$ be such that $y - Y^{(0)} = \sum_{i=1}^{i(y)-1}(\tau_2^{(i)} - \tau_1^{(i)}) + (\tau(y) - \tau_1^{(i(y))})$. From the implementation of Procedure(y), we have

$$\mathcal{I}^{B(y)} = \{[\tau(y), \tau_2^{(i(y))}], [\tau_1^{(i(y)+1)}, \tau_2^{(i(y)+1)}], \ldots, [\tau_1^{(m-1)}, \tau_2^{(m-1)}], [P^*, P^* + y]\}. \tag{8}$$

Equivalently, we also have $\mathcal{I}^{B(y)} = \{[\tau(y), \tau_2^{(i(y))}], h_{i(y)+1}, h_{i(y)+2}, \ldots, h_{m-1}, [P^*, P^* + y]\}$.

3 Two subroutines

In this section we introduce two subroutines that we will repeatedly use in our final algorithm. The first subroutine is implied in Wan et al. (2016), while we develop the second subroutine from an algorithm in Zhao and Yuan (2019).

3.1 The first subroutine

Let $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ be a job instance, where $d_1 \leq d_2 \leq \cdots \leq d_n$. Given $k \in \{1, 2, \ldots, n\}$, we use $\mathrm{TPE}(\mathcal{J}, k)$ to denote the problem to schedule the jobs of $\mathcal{J}$ on a single machine to minimize the total processing time of the early jobs (TPE) under the restriction that exactly k jobs are early. When no ambiguity will occur, we also use $\mathrm{TPE}(\mathcal{J}, k)$ to denote the optimal value of the problem. For the case where the problem is infeasible, we define $\mathrm{TPE}(\mathcal{J}, k) = +\infty$. According to Wan et al. (2016), the following algorithm, which we call "Subroutine $\mathrm{TPE}(\mathcal{J}, k)$", solves problem $\mathrm{TPE}(\mathcal{J}, k)$.

Subroutine TPE$(\mathcal{J}, k)$: For solving problem $\mathrm{TPE}(\mathcal{J}, k)$.

Step 1. From time 0, schedule the jobs one by one in the EDD order $J_1, J_2, \ldots, J_n$. When encountering a tardy job, delete the first longest job from the scheduled jobs. This procedure is repeated until one of the following two situations occurs:

(S1) exactly k early jobs are scheduled,

(S2) all the jobs are processed but fewer than k jobs are early.

Step 2. If (S2) occurs, the problem is infeasible. Then terminate the algorithm. If (S1) occurs, let $\mathcal{S}_l$ be the set of the k early jobs determined in Step 1, where l is the largest index of the early jobs, and go to Step 3.

Step 3. For $j = l + 1, l + 2, \ldots, n$, do the following iteratively:

Pick the first longest job $J_e \in \mathcal{S}_{j-1} \cup \{J_j\}$. Set $\mathcal{S}_j := (\mathcal{S}_{j-1} \cup \{J_j\}) \setminus \{J_e\}$ and $j := j+1$. Then repeat this procedure.

Step 4. From time 0, schedule the jobs in $\mathcal{S}_n$ in increasing order of their indices. Then terminate the algorithm.

By comparison, Subroutine $\mathrm{TPE}(\mathcal{J}, k)$ has the same time complexity as that of Moore's algorithm. Thus, Subroutine $\mathrm{TPE}(\mathcal{J}, k)$ runs in $O(n \log n)$ time. In fact, Wan et al. (2016) presented a procedure for single-machine scheduling with forbidden intervals that is more general than Subroutine $\mathrm{TPE}(\mathcal{J}, k)$. Then the following lemma is implied from Lemma 3.7 in Wan et al. (2016).

Lemma 3.1. *Subroutine TPE$(\mathcal{J}, k)$ solves problem TPE$(\mathcal{J}, k)$ in $O(n \log n)$ time.*

We also use the following lemma, which can be observed directly from the meaning of $\mathrm{TPE}(\mathcal{J}, k)$, for the subsequent analysis.

Lemma 3.2. *Suppose that $k' < k'' \leq n$ and $TPE(\mathcal{J}, k') < +\infty$. Then we have*

$$TPE(\mathcal{J}, k') < TPE(\mathcal{J}, k'').$$

Thus, $TPE(\mathcal{J}, k)$, when it is finite, is strictly increasing in k.

3.2 The second subroutine

Let $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$. Zhao and Yuan (2019) presented the following algorithm B-SRPT to solve problem $1|\text{pmtn}|\sum U_j$ on instance $\mathcal{J}$ and showed that the time complexity of BSRPT is $O(n \log n)$.

BSRPT: *Re-number the jobs such that $d_1 \leq d_2 \leq \cdots \leq d_n$. Starting with time $d_{\max} = d_n$, schedule the jobs preemptively and backwards using the strategy that, at any decision point τ (when a job is fully scheduled in the interval $[\tau, d_{\max}]$ or a smaller due date appears), schedules an uncompleted job with a due date having at least τ (if any) of the shortest unscheduled processing time. Finally, the unscheduled parts of the jobs at time 0 are re-scheduled at the end of the schedule.*

Let us consider the scheduling problem $(1, \mathcal{I})|\text{pmtn}|\sum U_j$ on instance $\mathcal{J}$, where $\mathcal{I} = \{h_k = [\tau_1^{(k)}, \tau_2^{(k)}] : k = 1, 2, \ldots, m\}$ is a set of m forbidden intervals as described in Section 2.2. For convenience, we use $J_j = (p_j, d_j)$ to indicate each job J_j of $\mathcal{J}$. In the preprocessing procedure, we re-number the jobs such that $d_1 \leq d_2 \leq \cdots \leq d_n$. By applying algorithm BSRPT to solve problem $1|\text{pmtn}|\sum U_j$ in Zhao and Yuan (2019), we use the following algorithm, denoted as "Subroutine FB($\mathcal{I}$)-BSRPT", to solve the scheduling problem $(1, \mathcal{I})|\text{pmtn}|\sum U_j$ on instance $\mathcal{J}$, where FB($\mathcal{I}$) means that $\mathcal{I}$ is the set of forbidden intervals.

Subroutine FB($\mathcal{I}$)-BSRPT: *Apply algorithm BSRPT on instance $\mathcal{J}$ in the idle time space $[0, +\infty) \setminus \mathcal{I}$ to schedule the jobs preemptively, with ties being broken by choosing the job with the largest index.*

The following theorem can be established based on the proof of Theorem 2.7 in Yuan and Lin (2005b).

Theorem 3.1. *Given $\mathcal{I}$ and $\mathcal{J}$, Subroutine FB($\mathcal{I}$)-BSRPT generates an optimal schedule for problem $(1, \mathcal{I})|pmtn|\sum U_j$ in $O(n \log n + m)$ time.*

Proof. For each $k \in \{1, 2, \ldots, m\}$, we set $H_k = |h_1| + |h_2| + \cdots + |h_{k-1}|$, which is the total length of the first $k-1$ forbidden intervals $h_1, h_2, \ldots, h_{k-1}$. Then we modify the due dates of the jobs in $\mathcal{J}$ in the following way:

$$\tilde{d}_j = \begin{cases} d_j, & \text{if } d_j \le \tau_1^{(1)}, \\ \tau_1^{(i)} - H_{i-1}, & \text{if } \tau_1^{(i)} \le d_j \le \tau_2^{(i)} \text{ for some } i \in \{1, 2, \ldots, m\}, \\ d_j - H_i, & \text{if } \tau_2^{(i)} \le d_j \le \tau_1^{(i+1)} \text{ for some } i \in \{1, 2, \ldots, m-1\}. \end{cases}$$

We define a new instance $\tilde{\mathcal{J}}$ by setting $\tilde{\mathcal{J}} = \{\tilde{J}_j = (p_j, \tilde{d}_j) : 1 \le j \le n\}$.

For each schedule σ for problem $(1, \mathcal{I})|\text{pmtn}|\sum U_j$ on instance $\mathcal{J}$, we define $\tilde{\sigma}$ as the schedule for problem $1|\text{pmtn}|\sum U_j$ on instance $\tilde{\mathcal{J}}$ that is obtained from σ by removing the m forbidden intervals of $\mathcal{I}$ and replacing $\mathcal{J}$ by $\tilde{\mathcal{J}}$. Based on the proof of Theorem 2.7 in Yuan and Lin (2005b), σ is an optimal schedule for problem $(1, \mathcal{I})|\text{pmtn}|\sum U_j$ on instance $\mathcal{J}$ if and only if $\tilde{\sigma}$ is an optimal schedule for problem $1|\text{pmtn}|\sum U_j$ on instance $\tilde{\mathcal{J}}$. By the definition of the modified due dates, for each $j \in \{1, 2, \ldots, n\}$, $d_j - \tilde{d}_j$ represents the length of the forbidden time space before time d_j. Thus, σ is a schedule generated by Subroutine FB($\mathcal{I}$)-BSRPT on instance $\mathcal{J}$ if and only if $\tilde{\sigma}$ is a schedule for $\mathcal{J}$ generated by Subroutine BSRPT on instance $\tilde{\mathcal{J}}$. From Zhao and Yuan (2019), algorithm BSRPT generates an optimal schedule for problem $1|\text{pmtn}|\sum U_j$. Consequently, Subroutine FB($\mathcal{I}$)-BSRPT generates an optimal schedule for problem $(1, \mathcal{I})|\text{pmtn}|\sum U_j$.

Finally, since we have a total of m forbidden intervals, the time complexity of Subroutine FB($\mathcal{I}$)-BSRPT is $O(m)$ plus the time complexity $O(n \log n)$ of BSRPT, i.e., $O(n \log n + m)$. The theorem follows. □

4 Pareto-optimal points

Recall the notation $Y^{(0)}$, $\mathcal{I}^{B(Y^{(0)})}$, and $\mathcal{I}^{B(y)}$ for $y \in (Y^{(0)}, P_B]$, which were defined in Sections 2.2 and 2.3 and will be repeatedly used in the sequel. We introduce some new notation in the following definition.

Definition 4.1. *Let $y \in (Y^{(0)}, P_B]$ and let $\tau \in (0, d^A_{n_A}]$.*

- *For each $i = 1, 2, \ldots, m-1$, we define $Y^{(i)} = Y^{(0)} + |h_1| + |h_2| + \cdots + |h_i|$ and $U^{(i)} = U(Y^{(i)})$. Then we have $Y^{(0)} < Y^{(1)} < \cdots < Y^{(m-1)} = P_B$ and $U^{(0)} \ge U^{(1)} \ge \cdots \ge U^{(m-1)}$. Note that $\tau(Y^{(i)}) = \tau_1^{(i+1)}$, which is the left endpoint of the interval h_{i+1}.*
- *We define*

$$\tau^*(y) = \begin{cases} \tau(y), & \text{if } y \notin \{Y^{(1)}, Y^{(2)}, \ldots, Y^{(m-1)}\}, \\ \tau_2^{(i)}, & \text{if } y = Y^{(i)} \text{ for some } i \in \{1, 2, \ldots, m-1\}. \end{cases}$$

Recall that $\tau(y)$ is the left endpoint of the first interval of $\mathcal{I}^{B(y)}$. Then we have

$$\tau^*(y) = \tau_1^{(i)} + (y - Y^{(i-1)}) \text{ if } y \in (Y^{(i-1)}, Y^{(i)}] \text{ for some } i. \tag{9}$$

• *m^* is the smallest index in $\{0,1,\ldots,m-1\}$ such that $U^{(m^*)} = U^{(m-1)}$.*

• *We use $\sigma^{A(y)}$ to denote the schedule of the A-jobs obtained by Subroutine FB($\mathcal{I}^{B(y)}$)-BSRPT for problem $(1,\mathcal{I}^{B(y)})|pmtn|\sum U_j^A$ on instance $\mathcal{J}^A$. Moreover, we use $\sigma^{(y)} = (\sigma^{A(y)},\sigma^{B(y)})$ to denote the schedule of $\mathcal{J}^A \cup \mathcal{J}^B$ in which the A-jobs are scheduled by $\sigma^{A(y)}$ and the B-jobs are scheduled by $\sigma^{B(y)}$. Then $\sigma^{(y)}$ is a y-optimal schedule.*

• *We use $\mathcal{J}^{A(y)}(\tau)$ to denote the parts of the A-jobs that are not scheduled in the time interval $[\tau, d^A_{n_A}]$ in the schedule $\sigma^{A(y)}$. For example, consider an A-job $J_j^A \in \mathcal{J}^A$. If J_j^A is fully scheduled in $[\tau, d^A_{n_A}]$, then $\mathcal{J}^{A(y)}(\tau)$ contains no part of J_j^A. If no part of J_j^A is scheduled in $[\tau, d^A_{n_A}]$, then $J_j^A \in \mathcal{J}^{A(y)}(\tau)$. If a part of J_j^A, denoted by $J_{j'}^A$, with $0 < p_{j'}^A < p_j^A$ is scheduled in $[\tau, d^A_{n_A}]$, then the remaining part of J_j^A, denoted by $J_{j''}^A = J_j^A \setminus J_{j'}^A$, is contained in $\mathcal{J}^{A(y)}(\tau)$. Note that $p_j^A = p_{j'}^A + p_{j''}^A$. Then we have $0 < p_{j''}^A < p_j^A$. In most cases, we just consider the parts set $\mathcal{J}^{A(y)}(\tau^*(y))$ in our discussion.*

• *For a schedule π and a time interval $[s,t]$, we use $\pi|_{[s,t]}$ to denote the restriction of π on $[s,t]$, and also call $\pi|_{[s,t]}$ the subschedule of π on $[s,t]$. We also use $\mathcal{J}^A(\pi|_{[s,t]})$ to denote the parts of the A-jobs in the subschedule $\pi|_{[s,t]}$. For the case where $s \geq t$, we define $\pi|_{[s,t]}$ and $\mathcal{J}^A(\pi|_{[s,t]})$ as an empty schedule and an empty set, respectively.*

• *We use $\nu(y,\tau)$ to denote the number of parts of $\mathcal{J}^{A(y)}(\tau)$ that are early in the subschedule $\sigma^{A(y)}|_{[0,\tau]}$ or, equivalently, that are fully scheduled in the interval $[0,\tau]$ in the schedule $\sigma^{A(y)}$.*

The following lemma establishes some useful properties for the notation introduced in Definition 4.1.

Lemma 4.1. *(i) For each $(u,y) \in \Omega(\mathcal{J}^A,\mathcal{J}^B)$, we have $Y^{(0)} \leq y \leq Y^{(m^*)}$ and $U^{(m^*)} \leq u \leq U^{(0)}$. Moreover, $(U^{(0)},Y^{(0)}) \in \Omega(\mathcal{J}^A,\mathcal{J}^B)$ and $(U^{(m^*)},y) \in \Omega(\mathcal{J}^A,\mathcal{J}^B)$ for some $y \in [Y^{(0)},Y^{(m^*)}]$.*

(ii) If $Y^{(0)} < y' < y'' \leq P_B$, then for every time point $\tau \geq \tau^*(y'')$, we have $\sigma^{A(y')}|_{[\tau,d^A_{n_A}]} = \sigma^{A(y'')}|_{[\tau,d^A_{n_A}]}$ and $\mathcal{J}^{A(y')}(\tau) = \mathcal{J}^{A(y'')}(\tau)$.

(iii) Suppose that $Y^{(i-1)} < y \leq Y^{(i)}$ for some $i \in \{1,2,\ldots,m-1\}$. Then for each $\tau \in [\tau^*(y),\tau_2^{(i)}]$, we have $\mathcal{J}^{A(y)}(\tau) = \mathcal{J}^{A(y)}(\tau_2^{(i)}) = \mathcal{J}^{A(Y^{(i)})}(\tau_2^{(i)})$ and $\nu(y,\tau) \leq \nu(Y^{(i)},\tau_2^{(i)})$.

(iv) For every $i \in \{1,2,\ldots,m-1\}$ and every two non-negative integers k_1 and k_2 with $k_1 < k_2$ and TPE$(\mathcal{J}^{A(Y^{(i)})},k_1) < +\infty$, we have TPE$(\mathcal{J}^{A(Y^{(i)})},k_1) <$ TPE$(\mathcal{J}^{A(Y^{(i)})},k_2)$.

(v) For every $y \in (Y^{(0)},P_B]$, we have TPE$(\mathcal{J}^{A(y)},\nu(y,\tau^*(y))) \leq \tau^*(y)$.

Proof. (i) follows by noting (from Lemma 2.2) that, if (u,y) is a Pareto-optimal point, then y is the minimum value in $[Y^{(0)},P_B] = [Y^{(0)},Y^{(m-1)}]$ such that $u = U(y)$.

(ii) follows from the implementation of Subroutines FB($\mathcal{I}^{B(y')}$)-BSRPT and FB($\mathcal{I}^{B(y'')}$)-BSRPT.

(iii) follows from (ii) and from the fact that $\tau^*(Y^{(i)}) = \tau_2^{(i)}$ and $[\tau^*(y),\tau_2^{(i)}]$ is a forbidden interval for the schedule $\sigma^{A(y)}$ of the A-jobs if $y < Y^{(i)}$.

(iv) follows from the fact that $\text{TPE}(\mathcal{J}^{A(Y^{(i)})}, k)$ (if it is a finite number) is strictly increasing in k since each part of $\mathcal{J}^{A(Y^{(i)})}$ (if it is non-empty) has a positive processing time.

(v) follows from the fact that we already have $\nu(y, \tau^*(y))$ early parts of $\mathcal{J}^{A(y)}(\tau^*(y))$ in the subschedule $\sigma^{A(y)}|_{[0,\tau^*(y)]}$. The lemma follows. $\square$

Now we assume that $\Omega(\mathcal{J}^A, \mathcal{J}^B) = \{(u_1, y_1), (u_2, y_2), \ldots, (u_K, y_K)\}$ such that $u_1 > u_2 > \cdots > u_K$ and $y_1 < y_2 < \cdots < y_K$. From Lemma 4.1(i), we have $(u_1, y_1) = (U^{(0)}, Y^{(0)})$, $u_K = U^{(m^*)}$, and $y_K \le Y^{(m^*)}$. If $m^* = 0$, then $(u_1, y_1) = (U^{(0)}, Y^{(0)})$ is the unique Pareto-optimal point, and we have nothing to do.

In general, we may suppose that $m^* \ge 1$. Our goal is to present a method to determine the points (u_i, y_i) for $i = 2, 3, \ldots, K$. The following lemma plays this role.

Lemma 4.2. *Let $i \in \{1, 2, \ldots, m^*\}$. Then we have the following three statements.*

(i) There is a point $(u_{z_i}, y_{z_i}) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$ such that $u_{z_i} = U^{(i)}$ and $y_{z_i} \le Y^{(i)}$.

(ii) If $U^{(i-1)} = U^{(i)}$, then there is no point $(u, y) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$ such that $y \in (Y^{(i-1)}, Y^{(i)}]$.

(iii) If $U^{(i-1)} > U^{(i)}$, then for each $u \in \{U^{(i)}, U^{(i)} + 1, \ldots, U^{(i-1)} - 1\}$, we have $(u, Y(u)) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$, where

$$Y(u) = \mathit{TPE}(\mathcal{J}^{A(Y^{(i-1)})}(\tau_1^{(i)}), \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u) - \tau_1^{(i)} + Y^{(i-1)}. \tag{10}$$

Proof. (i) and (ii) follow directly from Lemma 2.2. To prove (iii), note that $U^{(i)} \le u \le U^{(i-1)} - 1$. We first show the following inequalities

$$Y^{(i-1)} < Y(u) \le Y^{(i)}. \tag{11}$$

In fact, since $u < U^{(i-1)}$, we have $\nu(Y^{(i-1)}, \tau_1^{(i)}) < \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u$. If $\text{TPE}(\mathcal{J}^{A(Y^{(i-1)})}(\tau_1^{(i)}), \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u) \le \tau_1^{(i)}$, then there is a schedule of the parts of $\mathcal{J}^{A(Y^{(i-1)})}(\tau_1^{(i)})$ such that more than $\nu(Y^{(i-1)}, \tau_1^{(i)})$ early parts are scheduled in the interval $[0, \tau_1^{(i)}]$. This contradicts the optimality of schedule $\sigma^{A(Y^{(i-1)})}$ for $\mathcal{J}^{A(Y^{(i-1)})}(\tau_1^{(i)})$ in the interval $[0, \tau_1^{(i)}]$. Thus, we have

$$\text{TPE}(\mathcal{J}^{A(Y^{(i-1)})}(\tau_1^{(i)}), \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u) > \tau_1^{(i)}. \tag{12}$$

From (10) and (12), we obtain that $Y^{(i-1)} < Y(u)$. This proves the first inequality in (11).

From Lemma 4.1(ii), we obtain that $\sigma^{A(Y^{(i)})}|_{[\tau_2^{(i)}, d^A_{n_A}]} = \sigma^{A(Y^{(i-1)})}|_{[\tau_2^{(i)}, d^A_{n_A}]}$ and $\mathcal{J}^{A(Y^{(i)})}(\tau_2^{(i)}) = \mathcal{J}^{A(Y^{(i-1)})}(\tau_2^{(i)}) = \mathcal{J}^{A(Y^{(i-1)})}(\tau_1^{(i)})$. Since $u \ge U^{(i)}$, from the meaning of $\nu(Y^{(i)}, \tau_2^{(i)})$, we have $\nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u \le \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - U^{(i)} \le \nu(Y^{(i)}, \tau_2^{(i)})$. Thus, we have

$$\begin{aligned} & \mathrm{TPE}(\mathcal{J}^{A(Y^{(i-1)})}(\tau_1^{(i)}), \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u) \\ = \; & \mathrm{TPE}(\mathcal{J}^{A(Y^{(i)})}(\tau_1^{(i)}), \nu(Y^{(i-1)}, \tau_2^{(i)}) + U^{(i-1)} - u) \\ \le \; & \mathrm{TPE}(\mathcal{J}^{A(Y^{(i)})}(\tau_2^{(i)}), \nu(Y^{(i)}, \tau_2^{(i)})) \\ \le \; & \tau_2^{(i)}, \end{aligned} \tag{13}$$

where the two inequalities follow from Lemmas 3.2 and 4.1(v), respectively. From (10) and (13), we obtain that $Y(u) \le Y^{(i-1)} + \tau_2^{(i)} - \tau_1^{(i)} = Y^{(i)}$. This proves the second inequality in (11).

From (11), we have $\tau_1^{(i)} < \tau^*(Y(u)) \le \tau_2^{(i)}$. From (10) and from the definition of $\tau^*(y)$ for $y \in (Y^{(i-1)}, Y^{(i)}]$ in (9), we directly have

$$\tau^*(Y(u)) = \mathrm{TPE}(\mathcal{J}^{A(Y^{(i-1)})}(\tau_1^{(i)}), \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u). \tag{14}$$

From Lemma 4.1(ii), we have $\sigma^{A(Y^{(i)})}|_{[\tau^*(Y(u)), d_{n_A}^A]} = \sigma^{A(Y^{(i-1)})}|_{[\tau^*(Y(u)), d_{n_A}^A]}$. Thus, the number of early A-jobs scheduled in the interval $[\tau^*(Y(u)), d_{n_A}^A]$ in $\sigma^{A(Y(u))}$ is the same as that in $\sigma^{A(Y^{(i-1)})}$, which equals $n_A - U^{(i-1)} - \nu(Y^{(i-1)}, \tau_1^{(i)})$.

We now consider the number of parts of $\mathcal{J}^{A(Y(u))}(\tau^*(Y(u)))$ fully scheduled in the interval $[0, \tau^*(Y(u))]$ in $\sigma^{A(Y(u))}$. From Lemma 4.1(ii)-(iii), we have

$$\mathcal{J}^{A(Y^{(i-1)})}(\tau_1^{(i)}) = \mathcal{J}^{A(Y^{(i-1)})}(\tau^*(Y(u))) = \mathcal{J}^{A(Y(u))}(\tau^*(Y(u))).$$

It follows that

$$\begin{aligned} & \mathrm{TPE}(\mathcal{J}^{A(Y^{(i-1)})}(\tau_1^{(i)}), \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u) \\ = \; & \mathrm{TPE}(\mathcal{J}^{A(Y(u))}(\tau^*(Y(u))), \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u). \end{aligned}$$

By using (14), we conclude that

$$\tau^*(Y(u)) = \mathrm{TPE}(\mathcal{J}^{A(Y(u))}(\tau^*(Y(u))), \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u). \tag{15}$$

From Lemmas 3.1 and 3.2, and from (15), the number of early parts of $\mathcal{J}^{A(Y(u))}(\tau^*(Y(u)))$ scheduled in the interval $[0, \tau^*(Y(u))]$ in $\sigma^{A(Y(u))}$ is $\nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u$, i.e., $\nu(Y(u), \tau^*(Y(u))) = \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u$.

From the above analysis, the number of early jobs in $\sigma^{A(Y(u))}$ is

$$\begin{aligned} & n_A - U(Y(u)) \\ = \; & \nu(Y(u), \tau^*(Y(u))) + n_A - U^{(i-1)} - \nu(Y^{(i-1)}, \tau_1^{(i)}) \\ = \; & \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u + n_A - U^{(i-1)} - \nu(Y^{(i-1)}, \tau_1^{(i)}) \\ = \; & n_A - u. \end{aligned}$$

Consequently, we have

$$U(Y(u)) = u. \tag{16}$$

To complete the proof, it suffices to establish the following claim.

Claim 1. For every y with $Y^{(0)} < y < Y(u)$, $U(y) > u$.

From (16), we know that u is the optimal value of problem $1|\text{pmtn}|\sum U_j^A : \sum Y_j^B \leq Y(u)$. Thus, for every y with $Y^{(0)} < y < Y(u)$, $U(y) \geq u$. In the following we prove that $U(y) \neq u$.

If $y \leq Y^{(i-1)}$, then $U(y) \geq U(Y^{(i-1)}) = U^{(i-1)} > u$, as required.

Suppose in the following that $y > Y^{(i-1)}$. From (11), we have $Y^{(i-1)} < y < Y(u) \leq Y^{(i)}$. This means that

$$\tau_1^{(i)} < \tau^*(y) < \tau^*(Y(u)) \leq \tau_2^{(i)}. \tag{17}$$

From Lemma 4.1(ii), we have $\sigma^{A(y)}|_{[\tau^*(y),d^A_{n_A}]} = \sigma^{A(Y(u))}|_{[\tau^*(Y(u)),d^A_{n_A}]}$ and $\mathcal{J}^{A(y)}(\tau^*(y)) = \mathcal{J}^{A(Y(u))}(\tau^*(Y(u)))$. Thus, we can re-write (15) as

$$\tau^*(Y(u)) = \text{TPE}(\mathcal{J}^{A(y)}(\tau^*(y)), \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u). \tag{18}$$

Given the relation $\tau^*(y) < \tau^*(Y(u))$ in (15) and the relation in (18), from the definition of $\text{TPE}(\mathcal{J}, k)$ (or from Lemma 3.2), fewer than $\nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u$ parts of $\mathcal{J}^{A(y)}(\tau^*(y)) = \mathcal{J}^{A(Y(u))}(\tau^*(Y(u)))$ are early in schedule $\sigma^{A(y)}|_{[0,\tau^*(y)]}$. Since $\nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u$ is the number of early parts of $\mathcal{J}^{A(Y(u))}(\tau^*(Y(u)))$ scheduled in the interval $[0, \tau^*(Y(u))]$ in $\sigma^{A(Y(u))}$, we conclude that $U(y) > u$. This proves Claim 1.

From Lemma 2.2, (16) and Claim 1 enable us to conclude that $(u, Y(u))$ is a Pareto-optimal point. The result follows. □

As a direct consequence of Lemma 4.2, we have the following theorem.

Theorem 4.1. *For the instance $\mathcal{J}^A \cup \mathcal{J}^B$,*

$$\Omega(\mathcal{J}^A, \mathcal{J}^B) = \{(U^{(0)}, Y^{(0)})\} \cup \{(u, Y(u)) : u = U^{(m^*)}, U^{(m^*)} + 1, \ldots, U^{(0)} - 1\}, \tag{19}$$

where $Y(u)$ is defined in (10).

5 Polynomial-time algorithm

We are ready to present our algorithm to solve problem $1|\text{pmtn}|^{\#}(\sum U_j^A, \sum Y_j^B)$. Our algorithm is based on Theorem 4.1 and the analysis in previous sections. Note that we need not determine the value m^* before we generate the set $\Omega(\mathcal{J}^A, \mathcal{J}^B)$. Moreover, for each $y \in [Y^{(0)}, P_B]$, we use $\sigma^{(y)} = (\sigma^{A(y)}, \sigma^{B(y)})$ to denote the schedule for $\mathcal{J}^A \cup \mathcal{J}^B$ in which the subschedule for the A-jobs is $\sigma^{A(y)}$ and the subschedule for the B-jobs is $\sigma^{B(y)}$.

Algorithm 5.1. *For problem* $1|pmtn|^{\#}(\sum U_j^A, \sum Y_j^B)$.

Input: The job instance $\mathcal{J} = \mathcal{J}^A \cup \mathcal{J}^B$, where $\mathcal{J}^A = \{J_1^A, J_2^A, \ldots, J_{n_A}^A\}$ and $\mathcal{J}^B = \{J_1^B, J_2^B, \ldots, J_{n_B}^B\}$.

Preprocessing: Re-number the A-jobs in the EDD order such that $d_1^A \le d_2^A \le \cdots \le d_{n_A}^A$ with ties being broken by the LPT rule. Re-number the B-jobs in the EDD order with the B-jobs having the same due date being merged such that $d_1^B < d_2^B < \cdots < d_{n_B}^B$.

Step 1: Do the following:

(1.1) Generate schedule σ_0^B that schedules the B-jobs in the order $J_1^B \prec J_2^B \prec \cdots \prec J_{n_B}^B$ in the interval $[0, P_B]$ without idle times. Then calculate the value $Y^{(0)} = T_{\max}(\sigma_0^B)$.

(1.2) Invoke Procedure($Y^{(0)}$) to obtain schedule $\sigma^{B(Y^{(0)})}$ of the B-jobs. Determine the intervals $\mathcal{I}^{B(Y^{(0)})} = \{h_1, h_2, \ldots, h_m\}$ occupied by the B-jobs in $\sigma^{B(Y^{(0)})}$, where $h_l = [\tau_1^{(l)}, \tau_2^{(l)}]$ is the l-th interval, $l = 1, 2, \ldots, m$, as described in (6).

(1.3) Invoke Subroutine FB($\mathcal{I}^{B(Y^{(0)})}$)-BSRPT for problem $(1, \mathcal{I}^{B(Y^{(0)})})|\text{pmtn}|\sum U_j^A$ on instance $\mathcal{J}^A$ to obtain schedule $\sigma^{A(Y^{(0)})}$. Determine the value $U^{(0)} = \sum_{j=1}^{n_A} U_j^A(\sigma^{A(Y^{(0)})})$. Set $\sigma^{(Y^{(0)})} = (\sigma^{A(Y^{(0)})}, \sigma^{B(Y^{(0)})})$. Then $(U^{(0)}, Y^{(0)})$ is the first Pareto-optimal point and $\sigma^{(Y^{(0)})}$ is the corresponding Pareto-optimal schedule.

(1.4) Set $i := 1$ and set $\Omega(\mathcal{J}^A, \mathcal{J}^B) := \{(U^{(0)}, Y^{(0)})\}$. Go to Step 2.

Step 2: If $i = m$, then go to Step 5. If $i \le m - 1$, then go to Step 3.

Step 3: Do the following:

(3.1) Set $Y^{(i)} = Y^{(i-1)} + |h_i|$ and $\mathcal{I}^{B(Y^{(i)})} = \{h_{i+1}, h_{i+2}, \ldots, h_{m-1}, [P^*, P^* + Y^{(i)}]\}$.

(3.2) Invoke Subroutine FB($\mathcal{I}^{B(Y^{(i)})}$)-BSRPT for problem $(1, \mathcal{I}^{B(Y^{(i)})})|\text{pmtn}|\sum U_j^A$ on instance $\mathcal{J}^A$ to obtain schedule $\sigma^{A(Y^{(i)})}$. Determine the value $U^{(i)} = \sum_{j=1}^{n_A} U_j^A(\sigma^{A(Y^{(i)})})$.

(3.3) If $U^{(i-1)} = U^{(i)}$, then set $i := i + 1$ and go to Step 2. If $U^{(i-1)} > U^{(i)}$, then go to Step 4.

Step 4: Determine $\mathcal{J}^{A(Y^{(i-1)})}(\tau_1^{(i)})$ and $\nu(Y^{(i-1)}, \tau_1^{(i)})$ from schedule $\sigma^{A(Y^{(i-1)})}$.

For each $u \in \{U^{(i)}, U^{(i)}+1, \ldots, U^{(i-1)}-1\}$, invoke Subroutine TPE($\mathcal{J}^{A(Y^{(i-1)})}(\tau_1^{(i)}), \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u$) to determine the value TPE($\mathcal{J}^{A(Y^{(i-1)})}(\tau_1^{(i)}), \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u$). Calculate the value $Y(u)$ in the following way

$$Y(u) = \text{TPE}(\mathcal{J}^{A(Y^{(i-1)})}(\tau_1^{(i)}), \nu(Y^{(i-1)}, \tau_1^{(i)}) + U^{(i-1)} - u) - \tau_1^{(i)} + Y^{(i-1)}.$$

Generate schedule $\sigma^{(Y(u))} = (\sigma^{A(Y(u))}, \sigma^{B(Y(u))})$. Then $(u, Y(u))$ is a Pareto-optimal point and $\sigma^{(Y(u))}$ is the corresponding Pareto-optimal schedule.

Set $\Omega(\mathcal{J}^A, \mathcal{J}^B) := \Omega(\mathcal{J}^A, \mathcal{J}^B) \cup \{(u, Y(u)) : u = U^{(i)}, U^{(i)} + 1, \ldots, U^{(i-1)} - 1\}$ and $i := i + 1$. Go to Step 2.

Step 5: Output $\Omega(\mathcal{J}^A, \mathcal{J}^B)$ and, for each $(u, Y(u)) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$, the schedule $\sigma^{(Y(u))}$.

Remark: Note that the operations in Algorithm 5.1 can be re-arranged so that the repeated computations are reduced. But this does not affect the overall time complexity of the algorithm.

Theorem 5.1. *Algorithm 5.1 solves problem* $1|pmtn|^{\#}(\sum U_j^A, \sum Y_j^B)$ *in* $O(nn_A \log n_A + n_B \log n_B)$ *time.*

Proof. The correctness of Algorithm 5.1 is guaranteed by Lemma 2.3, Lemma 4.2, and Theorem 4.1. We next estimate the running time of Algorithm 5.1.

The Preprocessing procedure takes $O(n_A \log n_A + n_B \log n_B)$ time, which is dominated by the final time complexity. Moreover, the following facts are implied in the previous sections.

• $Y^{(0)}$ can be determined in $O(n_B \log n_B)$ time.

• Procedure($Y^{(0)}$) runs in $O(n_B)$ time. Then the forbidden intervals set $\mathcal{I}^{B(Y^{(0)})} = \{h_1, h_2, \ldots, h_m\}$ can be determined in $O(n_B)$ time.

• With $\mathcal{I}^{B(Y^{(0)})}$ being given, for each $y \in (Y^{(0)}, P_B]$, the items $\tau(y)$, $\tau^*(y)$, and $\mathcal{I}^{B(y)}$ can be determined in $O(n_B)$ time.

• With $\mathcal{I}^{B(Y^{(0)})}$ being given, Subroutine FB($\mathcal{I}^{B(Y^{(0)})}$)-BSRPT runs in $O(n_A \log n_A + n_B)$ time. As a result, the schedule $\sigma^{(Y^{(0)})} = (\sigma^{A(Y^{(0)})}, \sigma^{B(Y^{(0)})})$ can be obtained in $O(n_A \log n_A + n_B)$ time.

• With $\sigma^{A(Y^{(0)})}$ being given, for each $y \in (Y^{(0)}, P_B]$, $\tau(y)$ is the left endpoint of the first interval in $\mathcal{I}^{B(y)}$ and we have $\sigma^{A(y)}|_{[\tau(y), d^A_{n_A}]} = \sigma^{A(Y^{(0)})}|_{[\tau(y), d^A_{n_A}]}$. Thus, Subroutine FB($\mathcal{I}^{B(y)}$)-BSRPT can be implemented from time $\tau(y)$ for scheduling the parts of $\mathcal{J}^{A(y)}(\tau(y))$ in the interval $[0, \tau(y)]$ in $O(n_A \log n_A)$ time. Then the time complexity for generating the schedule $\sigma^{(y)} = (\sigma^{A(y)}, \sigma^{B(y)})$ is $O(n_A \log n_A)$ time.

• Given a set of parts of the A-jobs $\mathcal{J}'$ and a positive integer k, the value TPE($\mathcal{J}', k$) can be determined by Subroutine TPE($\mathcal{J}', k$) in $O(n_A \log n_A)$ time.

• The other operations in Algorithm 5.1 are non-dominating in the aspect of time complexity.

From the above facts, we can observe that Step 1 runs in $O(n_A \log n_A + n_B)$ time and, for each $i \in \{1, 2, \ldots, m\}$, Step 3 runs in $O(n_A \log n_A)$ time, and Step 4 runs in $O((U^{(i-1)} - U^{(i)}) n_A \log n_A)$ time. Note that $m \le n_B$ and $\sum_{i=1}^{m-1} (U^{(i-1)} - U^{(i)}) \le U^{(0)} \le n_A$. Then the time complexity of Algorithm 5.1 is

$$O(n_A \log n_A + n_B \log n_B) + O(n_B n_A \log n_A) + O(n_A n_A \log n_A),$$

which can be simplified as $O(nn_A \log n_A + n_B \log n_B)$. The result follows. $\square$

Now we give an instance to demonstrate the execution of Algorithm 5.1. Let $\mathcal{J} = \{J_1^A, J_2^A, \ldots, J_6^A, J_1^B, \ldots, J_4^B\}$ be the instance displayed in Table 1.

Table 1: The job instance $\mathcal{J}$

J_i^X	J_1^A	J_2^A	J_3^A	J_4^A	J_5^A	J_6^A	J_1^B	J_2^B	J_3^B	J_4^B
p_i^X	3	4	2	5	7	2	5	5	6	3
d_i^X	4	7	12	15	21	26	4	10	18	25

Note that $P^* = 1 + \max\{d_{n_A}^A, d_{n_B}^B\} = 1 + \max\{25, 26\} = 27$. Let $\Omega = \Omega(\mathcal{J}^A, \mathcal{J}^B)$. The key steps in applying Algorithm 5.1 to solve the instance are as follows:

(i) Generate the schedule $\sigma_0^B = (J_1^B, J_2^B, J_3^B, J_4^B)$ and calculate the Y-value $Y^{(0)} = T_{\max}(\sigma_0^B) = 1$. Then generate the schedule $\sigma^{B(Y^{(0)})}$, and the forbidden intervals $\mathcal{I}^{B(Y^{(0)})} = \{h_1, h_2, h_3, h_4, h_5\}$ is determined, where $h_1 = [0, 4]$, $h_2 = [5, 10]$, $h_3 = [12, 18]$, $h_4 = [22, 25]$, and $h_5 = [27, 28]$. Then, for each $y \in (Y^{(0)}, P_B] = (1, 19]$, $\sigma^{B(y)}$ and $\mathcal{I}^{B(y)}$ can be easily generated. The forbidden intervals of $\mathcal{I}^{B(Y^{(0)})} = \mathcal{I}^{B(1)}$ are displayed in Figure 1.

(ii) Generate the schedule $\sigma^{A(Y^{(0)})} = \sigma^{A(1)}$ and calculate the U-value $U^{(0)} = \sum U_j^A(\sigma^{A(1)}) = 4$. Then $\sigma^{(1)} = (\sigma^{A(1)}, \sigma^{B(1)})$ is a Pareto-optimal schedule corresponding to $(4, 1) \in \Omega$, as displayed in Figure 2.

(iii) Calculate $Y^{(1)} = Y^{(0)} + |h_1| = 5$, generate $\sigma^{A(5)}$, and calculate the U-value $U^{(1)} = U(5) = 3$. The schedule $\sigma^{A(5)}$ is displayed in Figure 3.

(iv) Now $U^{(0)} = 4 > 3 = U^{(1)}$. Calculate $Y(3) = 4$ and generate $\sigma^{A(4)}$. Then $\sigma^{(4)} = (\sigma^{A(4)}, \sigma^{B(4)})$ is a Pareto-optimal schedule corresponding to $(3, 4) \in \Omega$, as displayed in Figure 4.

(v) Calculate $Y^{(2)} = Y^{(1)} + |h_2| = 10$, generate $\sigma^{A(10)}$, and calculate the U-value $U^{(2)} = U(10) = 2$. The schedule $\sigma^{A(10)}$ is displayed in Figure 5.

(vi) Now $U^{(1)} = 3 > 2 = U^{(2)}$. Calculate $Y(2) = 7$ and generate $\sigma^{A(7)}$. Then $\sigma^{(7)} = (\sigma^{A(7)}, \sigma^{B(7)})$ is a Pareto-optimal schedule corresponding to $(2, 7) \in \Omega$, as displayed in Figure 6.

(vii) Calculate $Y^{(3)} = Y^{(2)} + |h_3| = 16$, generate $\sigma^{A(16)}$, and calculate the U-value $U^{(3)} = U(16) = 0$. The schedule $\sigma^{A(16)}$ is displayed in Figure 7.

(viii) Now $U^{(2)} = 2 > 0 = U^{(3)}$. For $u = U^{(2)} - 1 = 1$, we calculate $Y(u) = Y(1) = 11$, and generate $\sigma^{A(11)}$. Then $\sigma^{(11)} = (\sigma^{A(11)}, \sigma^{B(11)})$ is a Pareto-optimal schedule corresponding to $(1,11) \in \Omega$, as displayed in Figure 8.

(ix) For $u = U^{(3)} = 0$, calculate $Y(u) = Y(0) = 16$ and generate $\sigma^{A(16)}$. Then $\sigma^{(16)} = (\sigma^{A(16)}, \sigma^{B(16)})$ is a Pareto-optimal schedule corresponding to $(0,16) \in \Omega$, as displayed in Figure 9.

(x) Finally, we conclude that $\Omega = \{(4,1),(3,4),(2,7),(1,11),(0,16)\}$ and $\sigma^{(1)}, \sigma^{(4)}, \sigma^{(7)}, \sigma^{(11)}$, and $\sigma^{(16)}$ are the corresponding Pareto-optimal schedules.

Figure 1: The forbidden intervals of $\mathcal{I}^{B(1)}$.

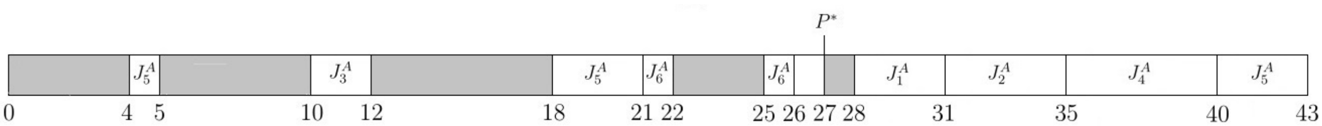

Figure 2: Schedule $\sigma^{(1)}$ corresponding to $(4,1) \in \Omega$.

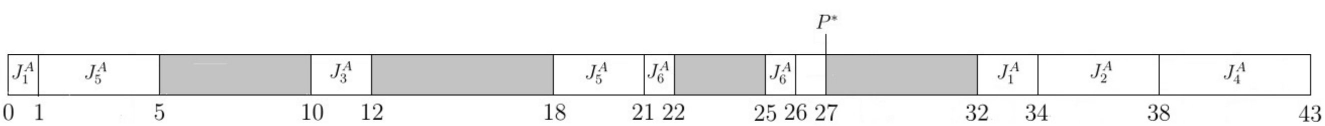

Figure 3: Schedule $\sigma^{A(5)}$ corresponding to $U^{(1)} = 3$.

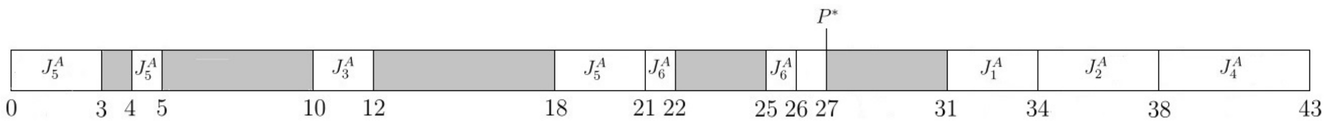

Figure 4: Schedule $\sigma^{(4)}$ corresponding to $(3,4) \in \Omega$.

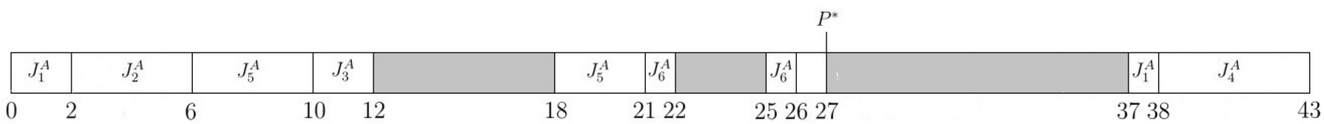

Figure 5: Schedule $\sigma^{A(10)}$ corresponding to $U^{(2)} = 2$.

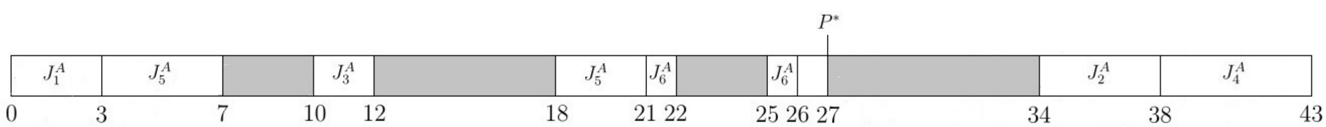

Figure 6: Schedule $\sigma^{(7)}$ corresponding to $(2,7) \in \Omega$.

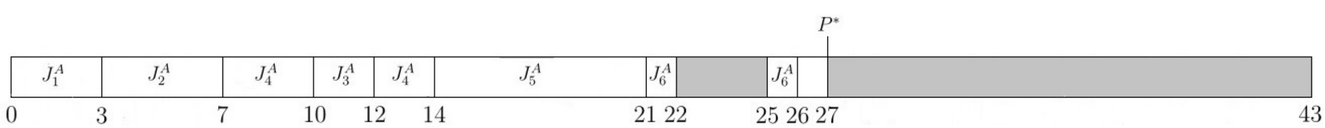

Figure 7: Schedule $\sigma^{A(16)}$ corresponding to $U^{(3)} = 0$.

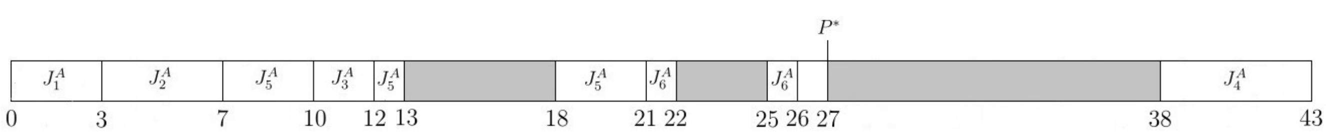

Figure 8: Schedule $\sigma^{(11)}$ corresponding to $(1,11) \in \Omega$.

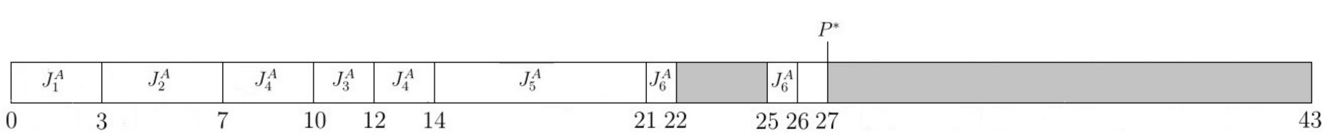

Figure 9: Schedule $\sigma^{(16)}$ corresponding to $(0,16) \in \Omega$.

Acknowledgment

This research was supported by NSFC (11671368) and NSFC (11771406).

References

Agnetis A, Mirchandani PB, Pacciarelli D, Pacifici A (2004) Scheduling problems with two competing agents. Operations Research 52:229-242

Agnetis A, Billaut JC, Gawiejnowicz S, Pacciarelli D, Soukhal A (2014) Multiagent scheduling: Models and algorithms. Springer, Berlin

Baker KR, Smith JC (2003) A multiple criterion model for machine scheduling. Journal of scheduling 6:7-16

Blazewicz J, Finke G (1987) Minimizing mean weighted execution time loss on identical and uniform processors. Information Processing Letters 24:259-263

Brucker P (2001) Scheduling Algorithms. Springer-Verlag, Berlin

Chen B, Potts CN, Woeginger GJ (1998) A review of machine scheduling: complexity, algorithms and approximability, in: D.Z. Du, P.M. Pardalos (Eds.), Handbook of Combinatorial Optimization, Kluwer Academic Publishers, Dordrecht, pp. 21-169.

Chen RB, Yuan JJ, Ng CT, Cheng TCE (2019) Single-machine scheduling with deadlines to minimize the total weighted late work. Naval Research Logistics 66:582-595

Cheng TCE, Ng CT, Yuan JJ (2006) Multi-agent scheduling on a single machine to minimize total weighted number of tardy jobs. Theoretical Computer Science 362:273-281

Cheng TCE, Ng CT, Yuan JJ (2008) Multi-agent scheduling on a single machine with max-form criteria. European Journal of Operational Research 188:603-609

Hariri AMA, Potts CN, Van Wassenhove LN (1995) Single machine scheduling to minimize total weighted late work. ORSA Journal on Computing 7:232-242

He RY, Yuan JJ (2019) Two-agent preemptive Pareto-scheduling to minimize late work and others. In submission

Lawler EL (1983) Recent results in the theory of machine scheduling, in: A. Bachem et al. (Eds.), Mathematical Programming-The State of the Art. Springer-Verlag, Berlin, pp. 202-234

Lee WC, Wang WJ, Shiau YR, Wu CC (2010) A single-machine scheduling problem with two-agent and deteriorating jobs. Applied Mathematical Modelling 34:3098-3107

Leung JYT, Pinedo M, Wan G (2010) Competitive Two-Agent Scheduling and Its Applications. Operations Research 58:458-469

Liu P, Gu M, Li G (2019) Two-agent scheduling on a single machine with release dates. Computers & Operations Research 111:35-42

Moore JM (1968) An n job, one machine sequencing algorithm for minimizing the number of late jobs. Management Science 15:102-109

Ng CT, Cheng TCE, Yuan JJ (2006) A note on the complexity of the problem of two-agent scheduling on a single machine. Journal of Combinatorial Optimization 12:387-394

Oron D, Shabtay D, Steiner G (2015) Single machine scheduling with two competing agents and equal jobs processing times. European Journal of Operational Research 244:86-99

Potts CN, Van Wassenhove LN (1991a) Single machine scheduling to minimize total late work. Operations Research 40:586-595

Potts CN, Van Wassenhove LN (1991b) Approximation algorithms for scheduling a single machine to minimize total late work. Operations Research Letters 11:261-266

Sterna M (2011) A survey of scheduling problems with late work criteria. Omega 39:120-129

T'kindt V, Billaut JC (2006) Multicriteria Scheduling: Theory, Models and Algorithms, second ed. Springer, Berlin

Wan L, Yuan JJ, Wei LJ (2016) Pareto optimization scheduling with two competing agents to minimize the number of tardy jobs and the maximum cost. Applied Mathematics and Computation 273:912-923

Yuan JJ (2018) Complexities of four problems on two-agent scheduling. Optimization Letters 12:763-780

Yuan JJ, Shang WP, Feng Q (2005a) A note on the scheduling with two families of jobs. Journal of scheduling 8:537-542

Yuan JJ, Lin YX (2005b) Single machine preemptive scheduling with fixed jobs to minimize tardiness related criteria. European Journal of Operational Research 164:851-855

Yuan JJ (2016) Complexities of some problems on multi-agent scheduling on a single machine. Journal of the Operations Research Society of China 4:379-384

Zhang XG, Wang Y (2017) Two-agent scheduling problems on a single-machine to minimize the total weighted late work. Journal of Combinatorial Optimization 33:945-955

Zhang Y, Yuan JJ (2019) A note on a two-agent scheduling problem related to the total weighted late work. Journal of Combinatorial Optimization 37:989-999

Zhao QL, Yuan JJ (2019) A note on single-machine scheduling to tradeoff between the number of tardy jobs and the start time of machine. Operations Research Letters 47:607-610