Two-Agent Preemptive Pareto-Scheduling to Minimize Late Work and Other Criteria

Ruyan He * and Jinjiang Yuan

School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China; yuanjj@zzu.edu.cn

* Correspondence: heruyan219@163.com or heruyan219@gs.zzu.edu.cn

Received: 12 August 2020; Accepted: 28 August 2020; Published: 5 September 2020

Abstract: In this paper, we consider three preemptive Pareto-scheduling problems with two competing agents on a single machine. In each problem, the objective function of agent A is the total completion time, the maximum lateness, or the total late work while the objective function of agent B is the total late work. For each problem, we provide a polynomial-time algorithm to characterize the trade-off curve of all Pareto-optimal points.

Keywords: two-agent; Pareto-scheduling; late work; trade-off curve; polynomial time

1. Introduction

In recent decades, scheduling with two competing agents and scheduling with late-work criterion have been two hot topics in scheduling research. However, research for the combination of the two topics has not been studied extensively. One reason for this phenomenon stems from the fact that the single-machine scheduling problem for minimizing the total late work is already NP-hard when preemption is not allowed. Given the polynomial solvability of the preemptive scheduling problem for minimizing the total late work, we study the single-machine two-agent preemptive Pareto-scheduling problems with the total late work being one of the criteria.

Problem Formulation: Consider two competing agents A and B . For each agent $X \in \{A, B\}$, let $\mathcal{J}^X = \{J_1^X, J_2^X, \dots, J_{n_X}^X\}$ be the set of jobs of agent X , where $\mathcal{J}^A \cap \mathcal{J}^B = \emptyset$ and the jobs in $\mathcal{J}^X$ are called the X -jobs. Each job J_j^X has a processing time $p_j^X > 0$ and a due date $d_j^X \geq 0$ which are integrally valued. The $n = n_A + n_B$ independent jobs in $\mathcal{J}^A \cup \mathcal{J}^B$ need to be preemptively processed on a single machine. Let $P_X = \sum_{j=1}^{n_X} p_j^X$ and $P = P_A + P_B$. All jobs considered in this paper are available at time zero. Our problems allow us to assume that the maximum due date of all jobs is at most P .

Let σ be a feasible schedule which assigns the jobs for processing in pieces in the interval $[0, +\infty)$. To enhance the flexibility of analysis, we allow the existence of idle times in a feasible schedule. The completion time of job J_j^X in σ is denoted as $C_j^X(\sigma)$. The late work of J_j^X in σ , denoted $Y_j^X(\sigma)$, is the amount of processing time of J_j^X after its due date d_j^X in σ . If $Y_j^X(\sigma) = 0$, J_j^X is called *early* in σ . If $0 < Y_j^X(\sigma) < p_j^X$, J_j^X is called *partially early* in σ . If $Y_j^X(\sigma) = p_j^X$, J_j^X is called *late* in σ .

The scheduling criteria related to our research are given by $\sum C_j^A = \sum_{j=1}^{n_A} C_j^A(\sigma)$ (the total completion time of the A -jobs under schedule σ), $L_{\max}^A = \max\{C_j^A(\sigma) - d_j^A : 1 \leq j \leq n_A\}$ (the maximum lateness of the A -jobs in schedule σ), and $\sum_j Y_j^X = \sum_{j=1}^{n_X} Y_j^X(\sigma)$ (the total late work of the X -jobs under schedule σ). Then the three Pareto-scheduling problems studied in this paper are given by

$$\begin{cases} 1|\text{pmtn}|^{\#}(\sum C_j^A, \sum Y_j^B), \\ 1|\text{pmtn}|^{\#}(L_{\max}^A, \sum Y_j^B), \\ 1|\text{pmtn}|^{\#}(\sum Y_j^A, \sum Y_j^B). \end{cases}$$

For each of the above three problems, we aim to find all the Pareto-optimal points of the problem and for each point a corresponding Pareto-optimal schedule. The formal definitions of Pareto-optimal point and Pareto-optimal schedule can be found in T'kindt and Billaut [1]. In this paper, the set of Pareto-optimal points forms a curve. Then we use the term *trade-off curve* to describe the set of Pareto-optimal points.

Literature Review: There is a huge amount of literature in scheduling with two competing agents and scheduling with late work criterion. Limited to the space of this paper, we only review some most related results.

Two-agent scheduling was first introduced by Agnetis et al. [2]. The authors proposed two important models for single-machine scheduling with two competing agents: constrained optimization model and Pareto optimization model. They studied nine problems arising from the different criteria combinations for the two agents and stated time complexity results for most of the resulting cases, where the scheduling criteria include the maximum regular cost function, the total (weighted) completion time and the number of tardy jobs. Early research on two-agent scheduling problems can also be found in Cheng et al. [3,4], Lee et al. [5], Leung et al. [6], and Ng et al. [7]. Agnetis et al. [8] applied the concept of price of fairness in resource allocation to two-agent single-machine scheduling problems, in which one agent aims at minimizing the total completion time, while the other agent wants to minimize the maximum tardiness with respect to a common due date. They further discussed the problem in which both agents wish to minimize the total completion time of their own jobs. Zhang et al. [9] studied the price of fairness in a two-agent single-machine scheduling problem in which both agents A and B want to minimize their own total completion time, and agent B has exactly two jobs.

There are few results on the two-agent scheduling problems with precedence constraints. Agnetis et al. [2] considered the two-agent problem with precedence constraints on single machine, i.e., $1|prec|f_{\max}^A : f_{\max}^B$ and solved this problem in polynomial time. Mor and Mosheiov [10] extended a classical single-machine scheduling problem, where the objective is to minimize maximum cost, given general job-dependent cost functions and general precedence constraints which was solved by the well-known Lawler's Algorithm. First, they allowed the option of job rejection. Then, they studied the more general setting of two competing agents, where job rejection is allowed either for one agent or for both. They showed that both extensions can be solved in polynomial time. Gao and Yuan [11] considered the Pareto-scheduling with two agents A and B for minimizing the total completion time of A -jobs and a maximum cost of B -jobs with precedence constraints. They showed that the problem can be solved in polynomial time. More recent results on two-agent scheduling problems can be found in Agnetis et al. [12], Liu et al. [13], Oron et al. [14], Perez-Gonzalez and Framinan [15], and Yuan [16,17].

Scheduling related to late-work criterion was first studied by Blazewicz and Finke [18]. Since then, several research groups have focused on this performance measure, obtaining a set of interesting results. Potts and Van Wassenhove [19] considered a single-machine scheduling problem where the goal is minimizing the total amount of late work. They showed that the problem is NP-hard and presented a pseudo-polynomial-time algorithm. Potts and Van Wassenhove [20] further proposed a branch-bound algorithm for the same problem and presented two fully polynomial-time approximation schemes with running times $O(\frac{n^2}{\epsilon})$ and $O(\frac{n^3}{\epsilon})$, respectively. Hariri et al. [21] considered a single-machine problem of minimizing the total weighted late work. They presented an $O(n \log n)$ algorithm for the preemptive total weighted late work problem. For papers that consider scheduling problem of minimizing other total late-work criteria, the reader may refer to the survey paper of Sterna [22].

Up to now, people have done little research about the combination of two-agent scheduling with late-work criterion. Wang et al. [23] addressed a two-agent scheduling problem where the objective is to minimize the total late work of the first agent, with the restriction that the maximum lateness of the second agent cannot exceed a given value. For small-scale problem instances, they established two pseudo-polynomial dynamic programming algorithms. For medium- to large-scale problem instances,

they presented a branch-and-bound algorithm. Zhang and Wang [24] presented a two-agent scheduling problem where the objective is to minimize the total weighted late work of agent A , while keeping the maximum cost of agent B cannot exceed a given bound U . They addressed the complexity of those problems, and presented the optimal polynomial-time algorithms or pseudo-polynomial-time algorithm to solve the scheduling problems, respectively. Zhang and Yuan [25] considered the same problem as above and further studied the three versions of the problem.

Our research also uses some results in the single-machine preemptive scheduling with forbidden intervals (or maintenance activities), i.e., $1, h_m | \text{pmtn} | f$, where “ $1, h_m$ ” means that there are m forbidden intervals on the single machine and “ f ” is the objective function to be minimized. Without reviewing this scheduling topic in detail, we only state two known results used in our discussion. Lee [26] showed that problem $1, h_m | \text{pmtn} | \sum C_j$ can be solved by the preemptive SPT rule in $O(m + n \log n)$ time and problem $1, h_m | \text{pmtn} | L_{\max}$ can be solved by the preemptive EDD rule in $O(m + n \log n)$ time.

Our Contributions: In Section 2, we introduce some notations and definitions and present several important lemmas. In Section 3, we show that the trade-off curve of problem $1 | \text{pmtn} |^\# (\sum C_j^A, \sum Y_j^B)$ can be determined in $O(nn_A n_B)$ time. In Section 4, we show that the trade-off curve of problem $1 | \text{pmtn} |^\# (L_{\max}^A, \sum Y_j^B)$ can be determined in $O(nn_A n_B)$ time. In Section 5, we show that the trade-off curve of problem $1 | \text{pmtn} |^\# (\sum Y_j^A, \sum Y_j^B)$ can be determined in $O(n \log n)$ time. Finally, some concluding remarks are given in Section 6.

2. Preliminaries

Let $\mathcal{J}^A \cup \mathcal{J}^B$ be the job instance to be preemptively scheduled on a single machine. The preemption assumption allows us to schedule each job in pieces. For a piece J_j^X of job J_j^X , we use p_j^X to denote the length (processing time) of J_j^X and use $C_j^X(\sigma)$ to denote the completion time of J_j^X in schedule σ .

In the following, we consider the Pareto-scheduling problem $1 | \text{pmtn} |^\# (f^A, \sum Y_j^B)$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$, where f^A is a regular objective function of the A -jobs or $f^A = \sum Y_j^A$. We use $\Omega(\mathcal{J}^A, \mathcal{J}^B)$ to denote the set of all Pareto-optimal points of this problem.

In a schedule σ of $\mathcal{J}^A \cup \mathcal{J}^B$, each B -job J_j^B is partitioned into two parts: the *early part* $J_j^{BE}(\sigma)$ and the *late part* $J_j^{BY}(\sigma)$, where $J_j^{BE}(\sigma)$ is processed before time d_j^B in σ and $J_j^{BY}(\sigma)$ is processed after time d_j^B in σ . Moreover, $p_j^{BE}(\sigma) = p_j^B - Y_j^B(\sigma)$ and $p_j^{BY}(\sigma) = Y_j^B(\sigma)$ are used to denote the lengths (processing times) of $J_j^{BE}(\sigma)$ and $J_j^{BY}(\sigma)$, respectively. A part of length 0 is called a *trivial part*. We allow the existence of trivial parts to enhance flexibility in analysis. Then we have

$$\sum_{j=1}^{n_B} Y_j^B(\sigma) = \sum_{j=1}^{n_B} p_j^{BY}(\sigma) = P_B - \sum_{j=1}^{n_B} p_j^{BE}(\sigma). \quad (1)$$

For convenience, we renumber the B -jobs such that

$$d_1^B \leq d_2^B \leq \dots \leq d_{n_B}^B \leq P \quad (2)$$

and keep this numbering throughout this paper. Let $\sigma_0^B = (J_1^B, J_2^B, \dots, J_{n_B}^B)$ which schedules the B -jobs in the EDD order described in (2). From Potts and Van Wassenhove [19], the optimal value of problem $1 | \text{pmtn} | \sum Y_j$ on instance $\mathcal{J}^B$ is given by $T_{\max}(\sigma_0^B)$. Then we have the following lemma.

Lemma 1. For each point $(f^*, Y^*) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$, we have $T_{\max}(\sigma_0^B) \leq Y^* \leq P_B$.

To make our analysis operational, we now consider an integer $Y^* \in [T_{\max}(\sigma_0^B), P_B]$ and present a procedure to schedule the B -jobs preemptively with a particular structure. This procedure imitates the algorithm in Hariri et al. [21] for solving problem $1 | \text{pmtn} | \sum w_j Y_j$.

Algorithm 1: For scheduling the B -jobs according to the value of Y^* .

Input: The B -jobs $\mathcal{J}^B$ with the EDD order in (2) and an integer $Y^* \in [T_{\max}(\sigma_0^B), P_B]$.

Step 1: Determine the minimum index $j^* \in \{1, 2, \dots, n_B\}$ such that $p_1^B + p_2^B + \dots + p_{j^*}^B \geq Y^*$. Then $p_1^B + p_2^B + \dots + p_{j^*-1}^B < Y^*$. Please note that if $Y^* = 0$, then we have $j^* = 1$. We call $J_{j^*}^B$ the *critical B -job* corresponding to Y^* .

Step 2: Decompose the critical B -job $J_{j^*}^B$ into two parts $J_{j^*}^{BE}$ and $J_{j^*}^{BY}$ such that

$$p_{j^*}^{BY} = Y^* - \sum_{j=1}^{j^*-1} p_j^B \text{ and } p_{j^*}^{BE} = p_{j^*}^B - p_{j^*}^{BY} = \sum_{j=1}^{j^*} p_j^B - Y^*.$$

We call $J_{j^*}^{BE}$ and $J_{j^*}^{BY}$ the *early part* and the *late part* of $J_{j^*}^B$, respectively, corresponding to Y^* .

Set $\mathcal{J}^{BE}(Y^*) = \{J_{j^*}^{BE}, J_{j^*+1}^B, J_{j^*+2}^B, \dots, J_{n_B}^B\}$ and $\mathcal{J}^{BY}(Y^*) = \{J_1^B, J_2^B, \dots, J_{j^*-1}^B, J_{j^*}^{BY}\}$.

Step 3: Generate a schedule $\sigma^{B(Y^*)}$ of the B -jobs $\mathcal{J}^B = \mathcal{J}^{BE}(Y^*) \cup \mathcal{J}^{BY}(Y^*)$ in the following way:

(3.1) From time $P^* := P + 1$, schedule the jobs (or pieces) in $\mathcal{J}^{BY}(Y^*)$ consecutively in the order $J_1^B, J_2^B, \dots, J_{j^*-1}^B, J_{j^*}^{BY}$.

(3.2) Schedule the jobs (or pieces) in $\mathcal{J}^{BE}(Y^*)$ by using the algorithm in Hariri et al. [21] for solving problem 1|pmtn| $\sum Y_j$ on instance $\mathcal{J}^{BE}(Y^*)$:

Beginning from time $d_{n_B}^B$, schedule the jobs (or pieces) in $\mathcal{J}^{BE}(Y^*)$ backwards in the order $J_{n_B}^B, J_{n_B-1}^B, \dots, J_{j^*+1}^B, J_{j^*}^{BE}$ such that each job (or piece) in $\mathcal{J}^{BE}(Y^*)$ is scheduled as late as possible subject to its due date.

Output: The schedule $\sigma^{B(Y^*)}$ of the B -jobs.

It can be observed that Procedure(Y^*) runs in $O(n)$ time. An objective function of the A -jobs, denoted f^A , is called *regular* if f^A is nondecreasing in the completion times of the A -jobs. Please note that $\sum C_j^A$ and $L_{\max}^A$ are regular, but $\sum Y_j^A$ is not regular since the preemptive assumption. The following lemma is critical in our discussion.

Lemma 2. Consider problem 1|pmtn| $^\#(f^A, \sum Y_j^B)$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$, where either $f^A = \sum Y_j^A$ or f^A is a regular objective function of the A -jobs. Assume that $(f^*, Y^*) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$ and let $\sigma^{B(Y^*)}$ be the schedule of $\mathcal{J}^B$ generated by Procedure(Y^*). Then there exists a Pareto-optimal schedule π corresponding to (f^*, Y^*) in which the B -jobs are scheduled in the same manner as that in $\sigma^{B(Y^*)}$. Such a Pareto-optimal schedule π is called a Y^* -standard schedule in the sequel.

Proof. Let σ be a Pareto-optimal schedule corresponding to (f^*, Y^*) such that $C_{\max}(\sigma)$ is as small as possible. Then no idle exists in σ , and so, $C_{\max}(\sigma) = P$.

The late parts of B -jobs can be scheduled arbitrarily late without affecting the objective values f^A and $\sum Y_j^B$. Thus, by shifting the late parts of B -jobs in σ , we obtain a new Pareto-optimal schedule σ_1 corresponding to (f^*, Y^*) such that the following property (P1) holds for σ_1 .

(P1) The late parts of B -jobs are scheduled consecutively in the interval $[P^*, P^* + Y^*]$ in an arbitrary order without idle time.

We next generate a Pareto-optimal schedule σ_2 corresponding to (f^*, Y^*) such that following property (P2) holds for σ_2 .

(P2) For every nontrivial early part $J_j^{BE}(\sigma_2)$ and every nontrivial late part $J_k^{BY}(\sigma_2)$ among the B -jobs, we have $j \geq k$.

If σ_1 has the property (P2), we just set $\sigma_2 = \sigma_1$. Otherwise, there are a nontrivial early part $J_j^{BE}(\sigma_1)$ and a nontrivial late part $J_k^{BY}(\sigma_1)$ among the B -jobs, such that $j < k$. From (2), we have $d_j^B \leq d_k^B$. Let $\delta = \min\{p_j^{BE}(\sigma_1), p_k^{BY}(\sigma_1)\}$. By exchanging an amount of length δ between $J_j^{BE}(\sigma_1)$ and $J_k^{BY}(\sigma_1)$ in σ_1 , we obtain a new schedule without changing the objective values but with improving in the direction we need. Repeating this procedure, we eventually obtain a new Pareto-optimal schedule σ_2 corresponding to (f^*, Y^*) such that both properties (P1) and (P2) hold for σ_2 .

Let σ_3 be the schedule obtained from σ_2 by rescheduling (if necessary) the early parts of B -jobs in the order $J_1^{BE}(\sigma) \prec J_2^{BE}(\sigma) \prec \dots \prec J_{n_B}^{BE}(\sigma)$. Since the EDD property described in (2), the early parts of B -jobs in σ_2 are also early in σ_3 . Then σ_3 is a Pareto-optimal schedule corresponding to (f^*, Y^*) such that properties (P1) and (P2), and additionally, the following property (P3), hold for σ_3 .

(P3) The early parts of B -jobs are scheduled in the order $J_1^{BE}(\sigma_3) \prec J_2^{BE}(\sigma_3) \prec \dots \prec J_{n_B}^{BE}(\sigma_3)$.

Please note that $\sum_{j=1}^{n_B} Y_j^B(\sigma_3) = \sum_{j=1}^{n_B} Y_j^B(\sigma^{B(Y^*)}) = Y^*$. Since σ_3 has the three properties (P1)–(P3), from Procedure (Y^*) for generating $\sigma^{B(Y^*)}$, we know that σ_3 and $\sigma^{B(Y^*)}$ have the same early parts and late parts of B -jobs. Then, in both schedules, the early parts of B -jobs are given by $\mathcal{J}^{BE}(Y^*) = \{J_{j^*}^{BE}, J_{j^*+1}^B, J_{j^*+2}^B, \dots, J_{n_B}^B\}$ and the late parts of B -jobs are given by $\mathcal{J}^{BY}(Y^*) = \{J_1^B, J_2^B, \dots, J_{j^*-1}^B, J_{j^*}^{BY}\}$, as defined in Step 2 of Procedure (Y^*) .

Now let π be the schedule obtained from σ_3 by the following two actions: (i) from time P^* , reschedule the late parts in $\mathcal{J}^{BY}(Y^*)$ consecutively in the order $J_1^B, J_2^B, \dots, J_{j^*-1}^B, J_{j^*}^{BY}$, and (ii) without changing the processing order of A -jobs and the processing order of the early parts of B -jobs, reschedule them such that the early parts of B -jobs are scheduled as late as possible subject to their due dates, and then, the A -jobs are scheduled as early as possible.

Clearly, in schedule π , the B -jobs are scheduled in the same manner as that in $\sigma^{B(Y^*)}$. Then we have $\sum_{j=1}^{n_B} Y_j^B(\pi) = Y^*$. From the construction of π , we have $f^A(\pi) \leq f^A(\sigma_3) = f^*$. The Pareto-optimality of (f^*, Y^*) further implies that $(f^A(\pi), \sum_{j=1}^{n_B} Y_j^B(\pi)) = (f^*, Y^*)$. Consequently, π is a required Pareto-optimal schedule corresponding to (f^*, Y^*) . The lemma follows. $\square$

From Lemmas 1 and 2, the Pareto-scheduling problem $1|pmtn|^\#(f^A, \sum Y_j^B)$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$ can be solved by the following general approach:

For each value $Y^* \in [T_{\max}(\sigma_0^B), P_B]$, run Procedure (Y^*) to obtain the schedule $\sigma^{B(Y^*)}$ of the B -jobs. Determine the intervals occupied by the B -jobs in $\sigma^{B(Y^*)}$ and regards these intervals as forbidden intervals. The intervals which are not occupied by the B -jobs in $\sigma^{B(Y^*)}$ is called the free-time intervals. Then solve problem $1, h_m|pmtn|f^A$ on instance $\mathcal{J}^A$ to obtain a Y^* -standard schedule.

The above approach cannot be implemented in polynomial time since it enumerates all the possible choices of Y^* . Therefore, in the next three sections, for $f^A \in \{\sum C_j^A, L_{\max}^A, \sum Y_j^A\}$, we will present polynomial-time algorithms, respectively, to characterize the trade-off curves.

To this end, we set $Y^{(0)} = T_{\max}(\sigma_0^B)$, and run Procedure $(Y^{(0)})$ to obtain the schedule $\sigma^{B(Y^{(0)})}$. Assume that the intervals occupied by the B -jobs are given by $h_1, h_2, \dots, h_m$, where $h_i = [\tau_1^{(i)}, \tau_2^{(i)}]$ is the i -th interval, $i = 1, 2, \dots, m$, such that

$$0 \leq \tau_1^{(1)} < \tau_2^{(1)} < \tau_1^{(2)} < \tau_2^{(2)} < \dots < \tau_1^{(m)} < \tau_2^{(m)}. \quad (3)$$

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

$$\tau_1^{(m)} = P^* \text{ and } \tau_2^{(m)} = P^* + Y^{(0)}. \quad (4)$$

For each $Y^* \in [Y^{(0)}, P_B]$, we define i^* to be the maximum index in $\{1, 2, \dots, m-1\}$ such that $Y^* - Y^{(0)} \geq \sum_{i=1}^{i^*-1} (\tau_2^{(i)} - \tau_1^{(i)})$ and let $\tau^* \in [\tau_1^{(i^*)}, \tau_2^{(i^*)})$ such that $Y^* - Y^{(0)} = \sum_{i=1}^{i^*-1} (\tau_2^{(i)} - \tau_1^{(i)}) + (\tau^* - \tau_1^{(i^*)})$.

From the implementation of Procedure(Y^*) again, the set of time intervals occupied by the B -jobs in schedule $\sigma^{B(Y^*)}$, denoted by $\mathcal{I}(\sigma^{B(Y^*)})$, is given by

$$\mathcal{I}(\sigma^{B(Y^*)}) = \{[\tau^*, \tau_2^{(i^*)}], [\tau_1^{(i^*+1)}, \tau_2^{(i^*+1)}], \dots, [\tau_1^{(m-1)}, \tau_2^{(m-1)}], [P^*, P^* + Y^*]\}. \quad (5)$$

We will write $h_{i^*} = [\tau^*, \tau_2^{(i^*)}]$, $h_i = [\tau_1^{(i)}, \tau_2^{(i)}]$ for $i = i^* + 1, i^* + 2, \dots, m - 1$, and $h_m = [P^*, P^* + Y^*]$.

The above discussion will help us to construct the trade-off curves easily.

3. The First Problem

In this section, we consider problem $1|pmtn|^\#(\sum C_j^A, \sum Y_j^B)$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$. By the job-exchanging argument, we can verify that the A -jobs must be scheduled in the SPT order in every Pareto-optimal schedule. Thus, in this section, we renumber the A -jobs by the SPT order such that $p_1^A \leq p_2^A \leq \dots \leq p_{n_A}^A$. Then we only consider the schedules in which the A -jobs are scheduled in the order $J_1^A \prec J_2^A \prec \dots \prec J_{n_A}^A$.

Given a point $(C^*, Y^*) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$, let σ be the Y^* -standard schedule of $\mathcal{J}^A \cup \mathcal{J}^B$. Then the set of forbidden intervals (occupied by the B -jobs) is given by (5) and the A -jobs are preemptively scheduled in the order $J_1^A \prec J_2^A \prec \dots \prec J_{n_A}^A$ from time 0 in the free-time intervals as early as possible. Thus, there are $m - i^* + 1$ forbidden intervals and the first forbidden interval in σ is given by $h_{i^*} = [\tau^*, \tau_2^{(i^*)}]$.

If $P_A \leq \tau^*$, then all the A -jobs are scheduled before the first forbidden interval h_{i^*} in σ . In this case, we have no further action.

In general, suppose that $P_A > \tau^*$. Then at least one A -job completes after h_{i^*} in σ . Let $J_{k(\sigma)}^A$ be the first A -job which completes after h_{i^*} in σ . Then, there are totally $n_A - k(\sigma) + 1$ A -jobs completing after h_{i^*} in σ .

For each index $j \in \{k(\sigma), k(\sigma) + 1, \dots, n_A\}$, we define $i[j]$ to be the interval index such that J_j^A completes after interval $h_{i[j]}$ and before interval $h_{i[j]+1}$ in σ , or equivalently, $\tau_2^{(i[j])} < C_j^A(\sigma) \leq \tau_1^{(i[j]+1)}$. We further define

$$e(\sigma) = \min\{C_j^A(\sigma) - \tau_2^{(i[j])} : j = k(\sigma), k(\sigma) + 1, \dots, n_A\}. \quad (6)$$

An A -job J_j^A with $j \in \{k(\sigma), k(\sigma) + 1, \dots, n_A\}$ is called a *crucial* A -job in σ if $C_j^A(\sigma) - \tau_2^{(i[j])} = e(\sigma)$. Please note that if J_j^A is a crucial A -job in σ , then the interval $[\tau_2^{(i[j])}, C_j^A(\sigma)]$ is fully occupied by job J_j^A in σ , implying that J_j^A is the first A -job completing after interval $h_{i[j]}$ in σ . In this case, we call interval $h_{i[j]}$ the *nearest* forbidden interval corresponding to crucial A -job J_j^A in σ .

Set $\delta_l(\sigma) = \tau_2^{(l)} - \tau_1^{(l)}(\sigma)$, $i^* \leq l \leq m$, to be the length of the forbidden interval h_l in σ . In particular, if $l = i^*$, then $\tau_1^{(l)}(\sigma) = \tau^*$. Let

$$\theta(\sigma) = \min\{e(\sigma), \delta_{i^*}(\sigma)\}. \quad (7)$$

Please note that when the schedule σ is given, the A -job index $k(\sigma)$ can be determined in $O(n_A)$ time, the interval indices $i[j]$ for $j \in \{k(\sigma), k(\sigma) + 1, \dots, n_A\}$ can be determined in $O(n_A)$ time. After that, the value $e(\sigma)$ defined in (6) can be determined in $O(n_A)$ time. Finally, the value $\theta(\sigma)$ can be determined by its definition in (7) in constant time. Then we have the following lemma.

Lemma 3. Given the Y^* -standard schedule σ in advance, the values $k(\sigma)$ and $\theta(\sigma)$ can be determined in $O(n_A)$ time.

For each $Y \in [Y^*, Y^* + \theta(\sigma)]$, let σ' be the Y -standard schedule. Then σ' is obtained from σ by shifting the first $Y - Y^*$ units of h_{i^*} to the last forbidden interval and then moving the A -jobs in $\{J_{k(\sigma)}^A, J_{k(\sigma)+1}^A, \dots, J_{n_A}^A\}$ left to eliminate the idle times accordingly. This means that $C_j^A(\sigma') \leq$

$C_j^A(\sigma) - (Y - Y^*)$ for $j \in \{k(\sigma), k(\sigma) + 1, \dots, n_A\}$. Assume that the total completion time of A -jobs in σ' is C . According to Lemma 2, (C, Y) is a Pareto-optimal point. In the following, we consider the trade-off curve between (C^*, Y^*) and (C, Y) . For convenience, point (C, Y) is simply called point Y .

We will show that the trade-off curve for $Y \in [Y^*, Y^* + \theta(\sigma))$ is a line segment. However, the point $Y^* + \theta(\sigma)$ may have the singularity.

Lemma 4. For each point $(C, Y) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$ with $Y \in [Y^*, Y^* + \theta(\sigma))$, we have $\frac{C - C^*}{Y - Y^*} = -(n_A - k(\sigma) + 1)$.

Proof. For each $Y \in [Y^*, Y^* + \theta(\sigma))$, we have $Y - Y^* < \theta(\sigma) \leq e(\sigma)$. When we change σ to σ' , no crucial A -jobs are moved left across their corresponding nearest forbidden intervals in σ' . As a result, compared with σ , each of the completion times of the $n_A - k(\sigma) + 1$ A -jobs in $\{J_{k(\sigma)}^A, J_{k(\sigma)+1}^A, \dots, J_{n_A}^A\}$ has decreased $Y - Y^*$ units in σ' . Thus, we have $C - C^* = -(n_A - k(\sigma) + 1)(Y - Y^*)$, as required. $\square$

Let $\mathcal{J}_c^A(\sigma)$ be the set of crucial A -jobs in σ . We use $\Delta(\sigma)$ to denote the total length of all the nearest forbidden intervals corresponding to the $\kappa(\sigma)$ crucial A -jobs in σ , i.e.,

$$\Delta(\sigma) = \sum \{\delta_{i[j]}(\sigma) : J_j^A \in \mathcal{J}_c^A(\sigma)\}.$$

The following lemma is only used to display the singularity of point $Y^* + \theta(\sigma)$.

Lemma 5. For the point $(C, Y) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$ with $Y = Y^* + \theta(\sigma)$, we have the following three statements.

- (i) If $e(\sigma) > \delta_{i^*}(\sigma)$, then $C = C^* - (n_A - k(\sigma) + 1)\theta(\sigma)$.
- (ii) If $e(\sigma) \leq \delta_{i^*}(\sigma)$ and the first crucial A -job is nearest to h_{i^*} in σ , then $C = C^* - \Delta(\sigma) - (n_A - k(\sigma))\theta(\sigma)$.
- (iii) If $e(\sigma) \leq \delta_{i^*}(\sigma)$ and no crucial A -job is nearest to h_{i^*} in σ , then $C = C^* - \Delta(\sigma) - (n_A - k(\sigma) + 1)\theta(\sigma)$.

Proof. When σ changes to σ' , each of the $n_A - k(\sigma) + 1$ A -jobs in $\{J_{k(\sigma)}^A, J_{k(\sigma)+1}^A, \dots, J_{n_A}^A\}$ is moved left $\theta(\sigma)$ units, and in the case that $e(\sigma) = \theta(\sigma)$, the crucial A -jobs are also moved left across their corresponding nearest forbidden intervals in σ' . Thus, we have

$$C = C^* - \Delta^*(\sigma') - (n_A - k(\sigma) + 1)\theta(\sigma), \quad (8)$$

where $\Delta^*(\sigma') = 0$ if $e(\sigma) > \theta(\sigma)$ and $\Delta^*(\sigma') = \sum \{\delta_{i[j]}(\sigma') : J_j^A \in \mathcal{J}_c^A(\sigma)\}$ if $e(\sigma) = \theta(\sigma)$. The key point is that $\delta_{i^*}(\sigma') = \delta_{i^*}(\sigma) - \theta(\sigma)$ and $\delta_l(\sigma') = \delta_l(\sigma)$ for $l = i^* + 1, i^* + 2, \dots, m$.

Under the assumption of (i), we have $\Delta^*(\sigma') = 0$. From (8), we have $C = C^* - (n_A - k(\sigma) + 1)\theta(\sigma)$.

Under the assumption of (ii), we have $\Delta^*(\sigma') = \Delta(\sigma) - \theta(\sigma)$. From (8), we have $C = C^* - \Delta(\sigma) - (n_A - k(\sigma))\theta(\sigma)$.

Under the assumption of (iii), we have $\Delta^*(\sigma') = \Delta(\sigma)$. From (8), we have $C = C^* - \Delta(\sigma) - (n_A - k(\sigma) + 1)\theta(\sigma)$. The lemma follows. $\square$

Theorem 1. Algorithm 2 generates the trade-off curve of $1|pmtn|^\#(\sum C_j^A, \sum Y_j^B)$ in $O(nn_A n_B)$ time.

Proof. The correctness of Algorithm 2 is guaranteed by Lemmas 2 and 4. We estimate the time complexity of the algorithm in the following.

The preprocessing procedure runs in $O(n_A \log n_A + n_B \log n_B)$ time. Each of Steps (1.1) and (1.2) runs in $O(n_B)$ time, so Step 1 runs in $O(n_B)$ time.

After Step 1, the algorithm has K iterations. In each iteration, either one forbidden interval is eliminated or at least one A -job is moved left across its corresponding nearest forbidden interval. Since $m \leq n_B$, we have $K = O(n_A n_B)$.

At each iteration, Step 2 runs in $O(n_A + m) = O(n)$ time. From Lemma 3, Step (3.1) runs in $O(n_A)$ time. Thus, each iteration runs in $O(n)$ time.

The above discussion establishes the $O(nn_A n_B)$ -time complexity of Algorithm 2. $\square$

Algorithm 2: Trade-off curve of problem $1|pmtn|^{\#}(\sum C_j^A, \sum Y_j^B)$.

Input: Instance $\mathcal{J}^A \cup \mathcal{J}^B$.

Preprocessing: Renumber the A -jobs such that $p_1^A \leq p_2^A \leq \dots \leq p_{n_A}^A$ and renumber the B -jobs such that $d_1^B \leq d_2^B \leq \dots \leq d_{n_B}^B$.

Step 1: Do the following:

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

(1.2) Run Procedure($Y^{(0)}$) to obtain the schedule $\sigma^{B(Y^{(0)})}$ of the B -jobs. Determine the intervals occupied by the B -jobs in $\sigma^{B(Y^{(0)})}$, say $h_1, h_2, \dots, h_m$, where $h_i = [\tau_1^{(i)}, \tau_2^{(i)}]$ is the i -th interval, $i = 1, 2, \dots, m$, as described in (3). Then regard $h_1, h_2, \dots, h_m$ as forbidden intervals which will be updated in the implementation of the algorithm. We take the convention that the forbidden intervals are just occupied by the B -jobs.

(1.3) Set $t_0 := 1$ and set $i := 0$.

Step 2: Do the following:

(2.1) Generate the $Y^{(i)}$ -standard schedule σ_i of $\mathcal{J}^A \cup \mathcal{J}^B$ in which $h_{t_i}, h_{t_i+1}, \dots, h_m$ are the forbidden intervals and the A -jobs are preemptively scheduled in the order $J_1^A \prec J_2^A \prec \dots \prec J_{n_A}^A$ as early as possible. Determine the value $C^{(i)} = \sum_{j=1}^{n_A} C_j^A(\sigma_i)$.

(2.2) If $J_{n_A}^A$ (and so, every A -job) is scheduled before the first forbidden interval h_{t_i} in σ_i , then set $K = i$ and go to Step 4. (In this case, we have obtained the whole trade-off curve.)

If $J_{n_A}^A$ completes after h_{t_i} in σ_i , then go to Step 3. (In this case, we use $J_{k(\sigma_i)}$ to denote the first A -job completing after h_{t_i} in σ_i .)

Step 3: Do the following:

(3.1) Calculate the values $k(\sigma_i)$ and $\theta(\sigma_i)$.

(3.2) Define a left closed right open segment $\mathcal{L}_i$ in the interval $[Y^{(i)}, Y^{(i)} + \theta(\sigma_i))$ by the following way:

$$\mathcal{L}_i : \frac{C - C^{(i)}}{Y - Y^{(i)}} = -(n_A - k(\sigma_i) + 1), Y \in [Y^{(i)}, Y^{(i)} + \theta(\sigma_i)).$$

(3.3) Set $Y^{(i+1)} := Y^{(i)} + \theta(\sigma_i)$ and $h_m := [P^*, P^* + Y^{(i+1)}]$. Moreover, if $\theta(\sigma_i) = \delta_{t_i}(\sigma_i)$, then set $t_{i+1} := t_i + 1$; and if $\theta(\sigma_i) < \delta_{t_i}(\sigma_i)$, then set $t_{i+1} := t_i$ and $h_{t_{i+1}} := [\tau_1^{(t_i)} + \theta(\sigma_i), \tau_2^{(t_i)}]$ (which is obtained from interval h_{t_i} by deleting the first $\theta(\sigma_i)$ units.)

(3.4) Set $i := i + 1$. Go to Step 2.

Step 4: Output the trade-off curve $\Omega(\mathcal{J}^A, \mathcal{J}^B) = \mathcal{L}_0 \cup \mathcal{L}_1 \cup \dots \cup \mathcal{L}_{K-1} \cup \{(C^{(K)}, Y^{(K)})\}$.

It can be observed that the total interruption time (i.e., the number of interruptions) of all the jobs in each schedule generated by Algorithm 2 is upper bounded by $1 + m \leq 1 + n_B$.

Let us consider the job instance I_1 displayed in Table 1. The trade-off curve of problem $1|pmtn|^{\#}(\sum C_j^A, \sum Y_j^B)$ on instance I_1 is shown in Figure 1.

Table 1. The job instance I_1 .

J_i^X	J_1^A	J_2^A	J_3^A	J_4^A	J_1^B	J_2^B	J_3^B
p_i^X	1	1	2	3	2	4	2
d_i^X	0	0	0	0	3	4	8

Please note that $P^* = P + 1 = 16$. Let $\Omega = \Omega(\mathcal{J}^A, \mathcal{J}^B)$. The key steps in applying Algorithm 2 to solve the instance I_1 are as follows:

- (i) Generate the schedule $\sigma_0^B = (J_1^B, J_2^B, J_3^B)$ and calculate the Y -value $Y^{(0)} = T_{\max}(\sigma_0^B) = 2$. Then generate the schedule $\sigma^{B(Y^{(0)})}$, and the forbidden interval set $\mathcal{I}^{B(Y^{(0)})} = \{h_1, h_2, h_3\}$ is determined, where $h_1 = [0, 4]$, $h_2 = [6, 8]$, and $h_3 = [16, 18]$. Then, for each $y \in (Y^{(0)}, P_B] = (2, 8]$, $\sigma^{B(y)}$ and $\mathcal{I}^{B(y)}$ can be easily generated.
- (ii) Generate the $Y^{(0)}$ -standard schedule σ_0 of $\mathcal{J}^A \cup \mathcal{J}^B$ in which h_1, h_2, h_3 are the forbidden intervals and the A -jobs are preemptively scheduled in the order $J_1^A \prec J_2^A \prec J_3^A \prec J_4^A$ as early as possible. Determine the value $C^{(0)} = \sum_{j=1}^{n_A} C_j^A(\sigma_0) = 34$. Then σ_0 is a Pareto-optimal schedule corresponding to $(34, 2) \in \Omega$.
- (iii) Calculate the values $k(\sigma_0) = 1$ and $\theta(\sigma_0) = 1$. Then line segment $\mathcal{L}_0$, which is the trade-off curve in the interval $[2, 3)$, satisfies $C = -4Y + 42$. Calculate $Y^{(1)} = Y^{(0)} + \theta(\sigma_0) = 3$ and the forbidden interval set $\mathcal{I}^{B(Y^{(1)})} = \{h_1, h_2, h_3\}$, where $h_1 = [1, 4]$, $h_2 = [6, 8]$, and $h_3 = [16, 19]$. Generate σ_1 , and calculate $C^{(1)} = \sum_{j=1}^{n_A} C_j^A(\sigma_1) = 27$. Then the intermediate point $(27, 3) \in \Omega$ is a jump discontinuity point and σ_1 is a Pareto-optimal schedule corresponding to $(27, 3)$.
- (iv) Calculate the values $k(\sigma_1) = 2$ and $\theta(\sigma_1) = 1$. Then the line segment $\mathcal{L}_1$, which is the trade-off curve in the interval $[3, 4)$, satisfies $C = -3Y + 36$. Calculate $Y^{(2)} = Y^{(1)} + \theta(\sigma_1) = 4$ and the forbidden interval set $\mathcal{I}^{B(Y^{(2)})} = \{h_1, h_2, h_3\}$, where $h_1 = [2, 4]$, $h_2 = [6, 8]$, and $h_3 = [16, 20]$. Generate σ_2 , and calculate $C^{(2)} = \sum_{j=1}^{n_A} C_j^A(\sigma_2) = 20$. Then the intermediate point $(20, 4) \in \Omega$ is a jump discontinuity point and σ_2 is a Pareto-optimal schedule corresponding to $(20, 4)$.
- (v) Calculate the values $k(\sigma_2) = 3$ and $\theta(\sigma_2) = 2$. Then the line segment $\mathcal{L}_2$, which is the trade-off curve in the interval $[4, 6)$, satisfies $C = -2Y + 28$. Calculate $Y^{(3)} = Y^{(2)} + \theta(\sigma_2) = 6$ and the forbidden interval set $\mathcal{I}^{B(Y^{(3)})} = \{h_2, h_3\}$, where $h_2 = [6, 8]$ and $h_3 = [16, 22]$. Generate σ_3 , and calculate $C^{(3)} = \sum_{j=1}^{n_A} C_j^A(\sigma_3) = 16$. Then the intermediate point $(16, 6) \in \Omega$ is a break point and σ_3 is a Pareto-optimal schedule corresponding to $(16, 6)$.
- (vi) Calculate the values $k(\sigma_3) = 4$ and $\theta(\sigma_3) = 1$. Then the line segment $\mathcal{L}_3$, which is the trade-off curve in the interval $[6, 7)$, satisfies $C = -Y + 22$. Calculate $Y^{(4)} = Y^{(3)} + \theta(\sigma_3) = 7$ and the forbidden interval set $\mathcal{I}^{B(Y^{(4)})} = \{h_2, h_3\}$, where $h_2 = [7, 8]$ and $h_3 = [16, 23]$. Generate σ_4 , and calculate $C^{(4)} = \sum_{j=1}^{n_A} C_j^A(\sigma_4) = 14$. Then the intermediate point $(14, 7) \in \Omega$ is a jump discontinuity point and σ_4 is a Pareto-optimal schedule corresponding to $(14, 7)$.
- (vii) Finally, we conclude that $\Omega = \{\mathcal{L}_0 \cup \mathcal{L}_1 \cup \mathcal{L}_2 \cup \mathcal{L}_3 \cup \{(14, 7)\}\}$ as displayed in Figure 1.

Figure 1. Trade-off curve.

4. The Second Problem

In this section, we consider problem $1|pmtn|^{\#}(L_{\max}^A, \sum Y_j^B)$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$. By the job-shifting argument, we can verify that for each Pareto-optimal point, there is a corresponding Pareto-optimal schedule in which the A -jobs are scheduled in the EDD order. Thus, in this section, we renumber the A -jobs by the EDD order such that $d_1^A \leq d_2^A \leq \dots \leq d_{n_A}^A$. Then we only consider the schedules in which the A -jobs are scheduled in the order $J_1^A \prec J_2^A \prec \dots \prec J_{n_A}^A$.

For a point $(L^*, Y^*) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$, let σ be a Y^* -standard schedule of $\mathcal{J}^A \cup \mathcal{J}^B$. Then the set of forbidden intervals (occupied by the B -jobs) is given by (5) and the A -jobs are preemptively scheduled in the order $J_1^A \prec J_2^A \prec \dots \prec J_{n_A}^A$ from time 0 in the free-time intervals. Thus, there are $m - i^* + 1$ forbidden intervals and the first forbidden interval in σ is given by $h_{i^*} = [\tau^*, \tau_2^{(i^*)}]$.

If $P_A \leq \tau^*$, then all the A -jobs are scheduled before the first forbidden interval h_{i^*} in σ . In this case, we have no further action.

In general, suppose that $P_A > \tau^*$. Then at least one A -job completing after h_{i^*} in σ . Let $J_{k(\sigma)}^A$ be the first A -job which completes after h_{i^*} in σ . Then, there are totally $n_A - k(\sigma) + 1$ A -jobs completing after h_{i^*} in σ .

Let $L^* = L_{\max}^A(\sigma)$ be the maximum lateness of A -jobs in σ . An A -job J_j^A with $k(\sigma) \leq j \leq n_A$ is called *critical* in σ if $L_j^A(\sigma) = L_{\max}^A(\sigma)$. Again, we use $\mathcal{J}_c^A(\sigma)$ to denote the set of all critical A -jobs. Then $\mathcal{J}_c^A(\sigma) = \{J_j^A : L_j^A(\sigma) = L^*, k(\sigma) \leq j \leq n_A\}$.

For each critical A -job J_j^A , we define $i[j]$ to be the interval index such that J_j^A completes after interval $h_{i[j]}$ and before interval $h_{i[j]+1}$ in σ , or equivalently, $\tau_2^{(i[j])} < C_j^A(\sigma) \leq \tau_1^{(i[j]+1)}$. Define

$$\lambda(\sigma) = \max\{C_j^A(\sigma) - \tau_2^{(i[j])} : J_j^A \in \mathcal{J}_c^A(\sigma)\}. \quad (9)$$

A critical A -job J_j^A is called a *desired* A -job in σ if $C_j^A(\sigma) - \tau_2^{(i[j])} = \lambda(\sigma)$. In this case, interval $h_{i[j]}$ is called the nearest forbidden interval corresponding to desired A -job J_j^A in σ . We further define

$$BL(\sigma) = \max\{L_j^A(\sigma) : C_j^A(\sigma) \leq \tau^*\}, \quad (10)$$

and

$$\Delta(\sigma) = L^* - BL(\sigma). \quad (11)$$

In the case that no A -job completes before interval h_{i^*} , we define $BL(\sigma) = -\infty$ and $\Delta(\sigma) = +\infty$. Moreover, we define

$$\vartheta(\sigma) = \min\{\min\{\lambda(\sigma), \delta_{i^*}(\sigma)\}, \Delta(\sigma)\}. \quad (12)$$

Please note that when the schedule σ is given, the values $C_j^A(\sigma)$ and $L_j^A(\sigma)$, $j = 1, 2, \dots, n_A$, and L^* can be determined in $O(n_A)$ time. Then the set $\mathcal{J}_c^A(\sigma)$ and the interval indices $i[j]$ for $J_j^A \in \mathcal{J}_c^A(\sigma)$ can be determined in $O(n_A)$ time. After that, the value $\lambda(\sigma)$, $BL(\sigma)$ and $\Delta(\sigma)$ can be determined in $O(n_A)$ time. Finally, the value $\vartheta(\sigma)$ can be determined by its definition in (12) in constant time. Then we have the following lemma.

Lemma 6. *Given the Y^* -standard schedule σ in advance, the values $\Delta(\sigma)$ and $\vartheta(\sigma)$ can be determined in $O(n_A)$ time.*

If $\Delta(\sigma) = 0$, then $BL(\sigma) = L^*$. In this case, (L^*, Y^*) is the last Pareto-optimal point.

Suppose that $\Delta(\sigma) > 0$. Then $\vartheta(\sigma) > 0$. For each $Y \in [Y^*, Y^* + \vartheta(\sigma)]$, let σ' be the Y -standard schedule. Then σ' is obtained from σ by shifting $Y - Y^*$ units of h_{i^*} to the last forbidden interval and then moving the A -jobs in $\{J_{k(\sigma)}^A, J_{k(\sigma)+1}^A, \dots, J_{n_A}^A\}$ left to eliminate the idle times accordingly. This means that $L_j^A(\sigma') \leq L_j^A(\sigma) - (Y - Y^*)$ for $j \in \{k(\sigma), k(\sigma) + 1, \dots, n_A\}$. Assume that the

maximum lateness of A -jobs in σ' is L . According to Lemma 2, (L, Y) is a Pareto-optimal point. In the following, we consider the trade-off curve between (L^*, Y^*) and (L, Y) . For convenience, point (L, Y) is simply called point Y .

We will show that the trade-off curve for $Y \in [Y^*, Y^* + \vartheta(\sigma))$ is a line segment. Discussion for the singularity of the point $Y^* + \vartheta(\sigma)$ will be omitted. This does not affect the characterization of the trade-off curve.

Lemma 7. Suppose that $\Delta(\sigma) > 0$. For each point $(L, Y) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$ with $Y \in [Y^*, Y^* + \vartheta(\sigma))$, we have $\frac{L-L^*}{Y-Y^*} = -1$.

Proof. For each $Y \in [Y^*, Y^* + \vartheta(\sigma))$, we have $Y - Y^* < \vartheta(\sigma)$. When we change σ to σ' , no desired A -jobs are moved left across their corresponding nearest forbidden intervals in σ' . As a result, compared with σ , each desired job must move forward $Y - Y^*$ units and the other A -jobs move forward at least $Y - Y^*$ units in σ' . Thus, the desired A -jobs in σ are also critical A -jobs in σ' . Then we have $\frac{L-L^*}{Y-Y^*} = -1$, as required. $\square$

Theorem 2. Algorithm 3 generates the trade-off curve of $1|\text{pmtn}|^\#(L_{\max}^A, \sum Y_j^B)$ in $O(nn_A n_B)$ time.

Algorithm 3: Trade-off curve of problem $1|\text{pmtn}|^\#(L_{\max}^A, \sum Y_j^B)$.

Input: Instance $\mathcal{J}^A \cup \mathcal{J}^B$.

Preprocessing: Renumber the A -jobs such that $d_1^A \leq d_2^A \leq \dots \leq d_{n_A}^A$ and renumber the B -jobs such that $d_1^B \leq d_2^B \leq \dots \leq d_{n_B}^B$.

Step 1: Do the following:

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

(1.2) Run Procedure($Y^{(0)}$) to obtain the schedule $\sigma^{B(Y^{(0)})}$ of the B -jobs. Determine the intervals occupied by the B -jobs in $\sigma^{B(Y^{(0)})}$, say $h_1, h_2, \dots, h_m$, where $h_i = [\tau_1^{(i)}, \tau_2^{(i)}]$ is the i -th interval, $i = 1, 2, \dots, m$, as described in (3). Then regard $h_1, h_2, \dots, h_m$ as forbidden intervals which will be updated in the implementation of the algorithm. We take the convention that the forbidden intervals are just occupied by the B -jobs.

(1.3) Set $t_0 := 1$ and set $i := 0$.

Step 2: Do the following:

(2.1) Generate the $Y^{(i)}$ -standard schedule σ_i of $\mathcal{J}^A \cup \mathcal{J}^B$ in which $h_{t_i}, h_{t_i+1}, \dots, h_m$ are the forbidden intervals and the A -jobs are preemptively scheduled in the order

$J_1^A \prec J_2^A \prec \dots \prec J_{n_A}^A$ as early as possible. Determine the values $L^{(i)} = L_{\max}^A(\sigma_i)$.

(2.2) Determine the value $\Delta(\sigma_i)$. Moreover, if $\Delta(\sigma_i) > 0$, determine the value $\vartheta(\sigma_i)$.

(2.3) If $\Delta(\sigma_i) = 0$, then set $K = i$ and go to Step 4. (In this case, we have obtained the whole trade-off curve.)

If $\Delta(\sigma_i) > 0$, then go to Step 3. (In this case, we have $\vartheta(\sigma_i) > 0$.)

Step 3: Define a left closed right open segment $\mathcal{L}_i$ in the interval $[Y^{(i)}, Y^{(i)} + \vartheta(\sigma_i))$ by the following way:

$$\mathcal{L}_i : \frac{L - L^{(i)}}{Y - Y^{(i)}} = -1, Y \in [Y^{(i)}, Y^{(i)} + \vartheta(\sigma_i)).$$

(3.3) Set $Y^{(i+1)} := Y^{(i)} + \vartheta(\sigma_i)$ and $h_m := [P^*, P^* + Y^{(i+1)}]$. Moreover, if $\vartheta(\sigma_i) = \delta_{t_i}(\sigma_i)$, then set $t_{i+1} := t_i + 1$; and if $\vartheta(\sigma_i) < \delta_{t_i}(\sigma_i)$, then set $t_{i+1} := t_i$ and $h_{t_{i+1}} := [\tau_1^{(t_i)} + \vartheta(\sigma_i), \tau_2^{(t_i)}]$ (which is obtained from interval h_{t_i} by deleting the first $\vartheta(\sigma_i)$ units.)

(3.4) Set $i := i + 1$. Go to Step 2.

Step 4: Output the trade-off curve $\Omega(\mathcal{J}^A, \mathcal{J}^B) = \mathcal{L}_0 \cup \mathcal{L}_1 \cup \dots \cup \mathcal{L}_{K-1} \cup \{(L^{(K)}, Y^{(K)})\}$.

Proof. Correctness of Algorithm 3 is guaranteed by Lemmas 2 and 7. The time complexity can be estimated by the similar way of Theorem 1 by putting Lemma 6 in discussion. $\square$

It can be observed that the total interruption time (i.e., the number of interruptions) of all the jobs in each schedule generated by Algorithm 3 is upper bounded by $1 + m \leq 1 + n_B$.

Let us consider the job instance I_2 displayed in Table 2. The trade-off curve of problem $1|pmtn|^{\#}(L_{\max}^A, \sum Y_j^B)$ on instance I_2 is shown in Figure 2.

Table 2. The instance I_2 .

J_i^X	J_1^A	J_2^A	J_3^A	J_1^B	J_2^B	J_3^B
p_i^X	4	2	3	5	2	2
d_i^X	4	7	11	3	7	12

Please note that $P^* = P + 1 = 19$. Let $\Omega = \Omega(\mathcal{J}^A, \mathcal{J}^B)$. The key steps in applying Algorithm 3 to solve the instance I_2 are as follows:

- (i) Generate the schedule $\sigma_0^B = (J_1^B, J_2^B, J_3^B)$ and calculate the Y -value $Y^{(0)} = T_{\max}(\sigma_0^B) = 2$. Then generate the schedule $\sigma^{B(Y^{(0)})}$, and the forbidden interval set $\mathcal{I}^{B(Y^{(0)})} = \{h_1, h_2, h_3, h_4\}$ is determined, where $h_1 = [0, 3]$, $h_2 = [5, 7]$, $h_3 = [10, 12]$, and $h_4 = [19, 21]$. Then, for each $y \in (Y^{(0)}, P_B] = (2, 9]$, $\sigma^{B(y)}$ and $\mathcal{I}^{B(y)}$ can be easily generated.
- (ii) Generate the $Y^{(0)}$ -standard schedule σ_0 of $\mathcal{J}^A \cup \mathcal{J}^B$ in which h_1, h_2, h_3, h_4 are the forbidden intervals and the A -jobs are preemptively scheduled in the order $J_1^A \prec J_2^A \prec J_3^A$ as early as possible. Determine the value $L^{(0)} = L_{\max}^A(\sigma_0) = 6$. Then σ_0 is a Pareto-optimal schedule corresponding to $(6, 2) \in \Omega$.
- (iii) Calculate the values $\Delta(\sigma_0) = 6 > 0$ and $\vartheta(\sigma_0) = 1$. Then the line segment $\mathcal{L}_0$, which is the trade-off curve in the interval $[2, 3)$, satisfies $L = -Y + 8$. Calculate $Y^{(1)} = Y^{(0)} + \vartheta(\sigma_0) = 3$ and the forbidden interval set $\mathcal{I}^{B(Y^{(1)})} = \{h_1, h_2, h_3, h_4\}$, where $h_1 = [1, 3]$, $h_2 = [5, 7]$, $h_3 = [10, 12]$, and $h_4 = [19, 22]$. Generate σ_1 , and calculate $L^{(1)} = L_{\max}^A(\sigma_1) = 4$. Then the intermediate point $(4, 3) \in \Omega$ is a jump discontinuity point and σ_1 is a Pareto-optimal schedule corresponding to $(4, 3)$.
- (iv) Calculate the values $\Delta(\sigma_1) = 4 > 0$ and $\vartheta(\sigma_1) = 2$. Then the line segment $\mathcal{L}_1$, which is the trade-off curve in the interval $[3, 5)$, satisfies $L = -Y + 7$. Calculate $Y^{(2)} = Y^{(1)} + \vartheta(\sigma_1) = 5$ and the forbidden interval set $\mathcal{I}^{B(Y^{(2)})} = \{h_2, h_3, h_4\}$, where $h_2 = [5, 7]$, $h_3 = [10, 12]$, and $h_4 = [19, 24]$. Generate σ_2 , and calculate $L^{(2)} = L_{\max}^A(\sigma_2) = 2$. Then the intermediate point $(2, 5) \in \Omega$ is a continuity point and σ_2 is a Pareto-optimal schedule corresponding to $(2, 5)$.
- (v) Calculate the values $\Delta(\sigma_2) = 2 > 0$ and $\vartheta(\sigma_2) = 1$. Then the line segment $\mathcal{L}_2$, which is the trade-off curve in the interval $[5, 6)$, satisfies $L = -Y + 7$. Calculate $Y^{(3)} = Y^{(2)} + \vartheta(\sigma_2) = 6$ and the forbidden interval set $\mathcal{I}^{B(Y^{(3)})} = \{h_2, h_3, h_4\}$, where $h_2 = [6, 7]$, $h_3 = [10, 12]$, and $h_4 = [19, 25]$. Generate σ_3 , and calculate $L^{(3)} = L_{\max}^A(\sigma_3) = 0$. Then the intermediate point $(0, 6) \in \Omega$ is a jump discontinuity point and σ_3 is a Pareto-optimal schedule corresponding to $(0, 6)$.
- (vi) Calculate the value $\Delta(\sigma_3) = 0$. Then, we conclude that $\Omega = \{\mathcal{L}_0 \cup \mathcal{L}_1 \cup \mathcal{L}_2 \cup \{(0, 6)\}\}$ as displayed in Figure 2.

Figure 2. Trade-off curve.

5. The Third Problem

In this section, we consider problem $1|pmtn|^{\#}(\sum Y_j^A, \sum Y_j^B)$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$. We renumber the A -jobs by the EDD order such that $d_1^A \leq d_2^A \leq \dots \leq d_{n_A}^A$.

For a point $(Y^A, Y^B) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$, a Y^B -standard schedule of $\mathcal{J}^A \cup \mathcal{J}^B$ corresponding to (Y^A, Y^B) can be obtained in the following way in $O(n)$ time: (i) Run Procedure(Y^B) to obtain the schedule $\sigma^{B(Y^B)}$ of the B -jobs, and (ii) run the algorithm in Hariri et al. [21] for solving problem $1|pmtn|\sum Y_j$ to schedule the A -jobs in the free-time intervals not occupied by the B -jobs in $\sigma^{B(Y^B)}$. Thus, we only need to consider the trade-off curve of problem $1|pmtn|^{\#}(\sum Y_j^A, \sum Y_j^B)$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$. We first establish a nice property for problem $1|pmtn|\sum Y_j$ in the following lemma.

Lemma 8. Let $\mathcal{J} = \{J_1, J_2, \dots, J_n\}$ be a job instance of problem $1|pmtn|\sum Y_j$. Let $\mathcal{U}$ be a subset of $\mathcal{J}$ such that there is a schedule of instance $\mathcal{J}$ such that all the jobs in $\mathcal{U}$ are early. Then there is an optimal schedule of problem $1|pmtn|\sum Y_j$ on instance $\mathcal{J}$ such that all the jobs in $\mathcal{U}$ are early.

Proof. We first prove the result for problem $1|pmtn|\sum Y_j$ without maintenance intervals by induction on $|\mathcal{U}|$. The result holds trivially if $|\mathcal{U}| = 0$.

Inductively, suppose that $|\mathcal{U}| = k \geq 1$, $\mathcal{U} = \{J_{j_1}, J_{j_2}, \dots, J_{j_k}\}$, $d_{j_1} \leq d_{j_2} \leq \dots \leq d_{j_k}$, and there is a feasible schedule of instance $\mathcal{J}$ such that all the jobs in $\mathcal{U}$ are early. Moreover, the result holds for every proper subset of $\mathcal{U}$ (the induction hypothesis).

Since $\mathcal{U} \setminus \{J_{j_k}\}$ is a proper subset of $\mathcal{U}$, from the induction hypothesis, there is an optimal schedule π of problem $1|pmtn|\sum Y_j$ on instance $\mathcal{J}$ such that all the $k-1$ jobs $J_{j_1}, J_{j_2}, \dots, J_{j_{k-1}}$ are early in π . Since all the jobs in $\mathcal{U}$ are early in some feasible schedule, we have $p_{j_1} + p_{j_2} + \dots + p_{j_k} \leq \max\{d_{j_1}, d_{j_2}, \dots, d_{j_k}\} = d_{j_k}$. This implies that all the $k-1$ jobs $J_{j_1}, J_{j_2}, \dots, J_{j_{k-1}}$ are completed by time d_{j_k} in π and at least p_{j_k} units of time in the interval $[0, d_{j_k}]$ are not occupied by the $k-1$ jobs $J_{j_1}, J_{j_2}, \dots, J_{j_{k-1}}$.

If $C_{j_k}(\pi) \leq d_{j_k}$, i.e., J_{j_k} is early in π , then π is a required optimal schedule.

Suppose in the following that $C_{j_k}(\pi) > d_{j_k}$. Then there is a certain index $i \in \{0, 1, \dots, p_{j_k}\}$ such that for job J_{j_k} , the first i unit pieces $J_{j_k,1}(\pi), J_{j_k,2}(\pi), \dots, J_{j_k,i}(\pi)$ are early in π and the last $p_{j_k} - i$ unit pieces $J_{j_k,i+1}(\pi), \dots, J_{j_k,p_{j_k}}(\pi)$ are late in π . Let $\mathcal{S}$ be the time space which consists of the last $p_{j_k} - i$ units of time in the interval $[0, d_{j_k}]$ that are not occupied by the $k-1$ jobs $J_{j_1}, J_{j_2}, \dots, J_{j_{k-1}}$ and the i unit pieces $J_{j_k,1}(\pi), J_{j_k,2}(\pi), \dots, J_{j_k,i}(\pi)$ of J_{j_k} . Let $\mathcal{T}$ be the time space which consists of the $p_{j_k} - i$

units of time that are occupied by the $p_{j_k} - i$ unit pieces $J_{j_k, i+1}(\pi), \dots, J_{j_k, p_{j_k}}(\pi)$ of J_{j_k} . Let σ be the schedule of $\mathcal{J}$ obtained from π by exchanging the subschedules in $\mathcal{S}$ and in $\mathcal{T}$. Then J_{j_k} is early in σ . Moreover, $\sum Y_j(\sigma) \leq \sum Y_j(\pi) - |\mathcal{T}| + |\mathcal{S}| = \sum Y_j(\pi)$, implying that σ is also optimal. Now all the jobs in $\mathcal{U} = \{J_{j_1}, J_{j_2}, \dots, J_{j_k}\}$ are early in σ . Consequently, σ is an optimal schedule of problem $1|pmtn|\sum Y_j$ on instance $\mathcal{J}$ such that all the jobs in $\mathcal{U}$ are early. The result follows by the induction principle. $\square$

We next use Lemma 8 to prove the following useful lemma.

Lemma 9. Let $\mathcal{J} = \{J_1, J_2, \dots, J_n\}$ be a job instance of problem $1|pmtn|\sum Y_j$. Let π be a schedule of the jobs in $\mathcal{J}$. Then there is an optimal schedule σ of problem $1|pmtn|\sum Y_j$ on instance $\mathcal{J}$ such that $Y_j(\sigma) \leq Y_j(\pi)$ for $j = 1, 2, \dots, n$.

Proof. For each $j \in \{1, 2, \dots, n\}$, we partition J_j into two parts $J_{j'}$ and $J_{j''}$ such that $p_{j'} = p_j - Y_j(\pi)$ is the early work of J_j in π , $p_{j''} = Y_j(\pi)$ is the late work of J_j in π , and $d_{j'} = d_{j''} = d_j$. Let $\mathcal{J}' = \{J_{j'}, J_{j''} : j = 1, 2, \dots, n\}$. Let π' be the schedule of $\mathcal{J}'$ which is obtained from π by just regarding the early part of J_j in π as job $J_{j'}$ and regarding the late part of J_j in π as job $J_{j''}$. Then all the jobs in $\{J_{j'} : j = 1, 2, \dots, n\}$ are early in π' . According to Lemma 8, there is an optimal schedule σ of problem $1|pmtn|\sum Y_j$ on instance $\mathcal{J}'$ such that all the jobs in $\{J_{j'} : j = 1, 2, \dots, n\}$ are early. Since the preemptive assumption, the two instances $\mathcal{J}$ and $\mathcal{J}'$ have no essential difference for problem $1|pmtn|\sum Y_j$, σ is an optimal schedule of problem $1|pmtn|\sum Y_j$ on instance $\mathcal{J}$. The result follows by noting that $Y_j(\sigma) = Y_{j''}(\sigma) \leq p_{j''} = Y_j(\pi)$ for $j = 1, 2, \dots, n$. $\square$

Let Y^{AB} be the optimal value of problem $1|pmtn|\sum Y_j$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$. We have the following lemma.

Lemma 10. For each point $(Y^A, Y^B) \in \Omega(\mathcal{J}^A, \mathcal{J}^B)$, we have $Y^A + Y^B = Y^{AB}$.

Proof. Let σ be a Pareto-optimal schedule of problem $1|pmtn|^\#(\sum Y_j^A, \sum Y_j^B)$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$ such that $\sum Y_j^A(\sigma) = Y^A$ and $\sum Y_j^B(\sigma) = Y^B$. From Lemma 9, there is an optimal schedule σ' of $1|pmtn|\sum Y_j$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$ such that $\sum Y_j^A(\sigma') \leq \sum Y_j^A(\sigma)$ and $\sum Y_j^B(\sigma') \leq \sum Y_j^B(\sigma)$. The optimality of σ' implies that $\sum Y_j(\sigma') = Y^{AB}$. From the property of Pareto-optimal point, we can obtain that $\sum Y_j^A(\sigma') = \sum Y_j^A(\sigma)$ and $\sum Y_j^B(\sigma') = \sum Y_j^B(\sigma)$. Thus, we have $Y^A + Y^B = Y^{AB}$. The result follows. $\square$

Theorem 3. The trade-off curve of problem $1|pmtn|^\#(\sum Y_j^A, \sum Y_j^B)$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$ can be determined in $O(n \log n)$ time.

Proof. Let $Y_A^{(0)}$ be the optimal value of problem $1|pmtn|Y_j^A$ on instance $\mathcal{J}^A$. Let $Y_B^{(0)}$ be the optimal value of problem $1|pmtn|Y_j^B$ on instance $\mathcal{J}^B$. Recall that Y^{AB} is the optimal value of problem $1|pmtn|\sum Y_j$ on instance $\mathcal{J}^A \cup \mathcal{J}^B$. From Hariri et al. [21], $Y_A^{(0)}$, $Y_B^{(0)}$, and Y^{AB} can be determined in $O(n_A \log n_A)$ time, $O(n_B \log n_B)$ time, and $O(n \log n)$ time, respectively.

Please note that $Y_A^{(0)}$ is the minimum total late work of A-jobs among all Pareto-optimal points and $Y_B^{(0)}$ is the minimum total late work of B-jobs among all Pareto-optimal points. Thus, from Lemma 10, the trade-off curve is the line segment

$$Y^A + Y^B = Y^{AB}, Y_B^{(0)} \leq Y^B \leq Y^{AB} - Y_A^{(0)},$$

connecting point $(Y^{AB} - Y_B^{(0)}, Y_B^{(0)})$ to point $(Y_A^{(0)}, Y^{AB} - Y_A^{(0)})$. So, the overall complexity to obtain the trade-off curve is given by $O(n \log n)$. $\square$

It can be observed that the total interruption time (i.e., the number of interruptions) of all the jobs in each Pareto-optimal schedule is upper bounded by $\max\{n_A, n_B\} + 2$.

Let us consider the job instance I_3 displayed in Table 3. The trade-off curve of problem $1|pmtn|^{\#}(\sum Y_j^A, \sum Y_j^B)$ on instance I_3 is shown in Figure 3.

Table 3. The instance I_3 .

J_i^X	J_1^A	J_2^A	J_3^A	J_4^A	J_5^A	J_1^B	J_2^B	J_3^B
p_i^X	4	2	3	2	2	3	3	3
d_i^X	3	4	7	11	14	2	8	13

Please note that $P^* = P + 1 = 23$. Let $\Omega = \Omega(\mathcal{J}^A, \mathcal{J}^B)$. The key steps to solve the instance I_3 are as follows:

- Generate the schedule $\sigma_0^B = (J_1^B, J_2^B, J_3^B)$ and calculate the Y^B -value $Y^{(0)} = T_{\max}(\sigma_0^B) = 1$. Then generate the schedule $\sigma^{B(Y^{(0)})}$, and the forbidden interval set $\mathcal{I}^{B(Y^{(0)})} = \{h_1, h_2, h_3, h_4\}$ is determined, where $h_1 = [0, 2]$, $h_2 = [5, 8]$, $h_3 = [10, 13]$, and $h_4 = [23, 24]$. Then, for each $y \in (Y^{(0)}, P_B] = (1, 9]$, $\sigma^{B(y)}$ and $\mathcal{I}^{B(y)}$ can be easily generated.
- Generate the $Y^{(0)}$ -standard schedule σ_0 of $\mathcal{J}^A \cup \mathcal{J}^B$ in which h_1, h_2, h_3, h_4 are the forbidden intervals and the A -jobs are preemptively scheduled by running the algorithm in Hariri et al. [21] for solving problem $1|pmtn|\sum Y_j$ in the free-time intervals not occupied by the B -jobs in $\sigma^{B(Y^{(0)})}$. Determine the value $Y^A(\sigma_0) = \sum Y_j^A(\sigma_0) = 7$. Then σ_0 is a Pareto-optimal schedule corresponding to $(7, 1) \in \Omega$.
- By using the same method as (i) and (ii), we schedule A -jobs first. Generate the schedule $\sigma^{A(Y^{(0)})}$, where $Y^{(0)} = T_{\max}(\sigma_0^A) = 2$. Generate the schedule σ_1 of $\mathcal{J}^A \cup \mathcal{J}^B$ in which the B -jobs are preemptively scheduled by running the algorithm in Hariri et al. [21] for solving problem $1|pmtn|\sum Y_j$ in the free-time intervals not occupied by the A -jobs in $\sigma^{A(Y^{(0)})}$. Then σ_1 is a Pareto-optimal schedule corresponding to $(2, 6) \in \Omega$. From Lemma 10, σ_1 is the final Pareto-optimal schedule. Then the trade-off curve is just the line segment in the interval $[1, 6]$, which satisfies $Y^A = -Y^B + 8$ as displayed in Figure 3.

Figure 3. Trade-off curve.

6. Conclusions

This paper considers three preemptive Pareto-scheduling problems with two competing agents on a single machine. Two agents compete to perform their respective jobs on a common single machine and each agent has his own criterion to optimize. In each problem, the goal of agent A is to minimize the total completion time, the maximum lateness, or the total late work while agent B wants to minimize the total late work. For each problem, we provide a polynomial-time algorithm to characterize the trade-off curve of all Pareto-optimal points.

Late-work criterion can be met in all cases where the penalty imposed on a solution depends on the number of tardy units of jobs performed in a system. For example, in production planning where the manufacturer is concerned with minimizing any order delays which cause financial loss, in control systems where the accuracy of control procedures depends on the amount of information provided as their input, in agriculture where performance measures based on due dates, and so on. In the case where two criteria need to be minimized, the trade-off curve results an ideal solution. Once the trade-off curve is characterized, decision makers can make decisions as needed.

For the future research, the trade-off curve of the problem $1|pmtn|^{\#}(\sum C_j^A, \sum w_j Y_j^B)$ or $1|pmtn|^{\#}(L_{\max}^A, \sum w_j Y_j^B)$ is worthy of study. Since the existence of precedence constraints on scheduling problems reflects real-life problems, it is also worthy to study the two-agent problems with precedence constraints. Another interesting future research direction is to investigate fairness issues when the total late work is one of the criteria in two-agent scheduling problems.

Author Contributions: Conceptualization, methodology, and writing—original manuscript: R.H.; project management, supervision, and writing—review: J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grant numbers 11671368 and 11771406.

Acknowledgments: The authors would like to thank the Associate Editor and two anonymous referees for their constructive comments and helpful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. T'kindt, V.; Billaut, J.C. *Multicriteria Scheduling: Theory, Models and Algorithms*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2006.
2. Agnetis, A.; Mirchandani, P.B.; Pacciarelli, D.; Pacifici, A. Scheduling problems with two competing agents. *Oper. Res.* **2004**, *52*, 229–242. [\[CrossRef\]](#)
3. Cheng, T.C.E.; Ng, C.T.; Yuan, J.J. Multi-agent scheduling on a single machine to minimize total weighted number of tardy jobs. *Theor. Comput. Sci.* **2006**, *362*, 273–281. [\[CrossRef\]](#)
4. Cheng, T.C.E.; Ng, C.T.; Yuan, J.J. Multi-agent scheduling on a single machine with max-form criteria. *Eur. J. Oper. Res.* **2008**, *188*, 603–609. [\[CrossRef\]](#)
5. Lee, W.C.; Wang, W.J.; Shiau, Y.R.; Wu, C.C. A single-machine scheduling problem with two-agent and deteriorating jobs. *Appl. Math. Model.* **2010**, *34*, 3098–3107. [\[CrossRef\]](#)
6. Leung, J.Y.T.; Pinedo, M.; Wan, G.H. Competitive two-agent scheduling and its applications. *Oper. Res.* **2010**, *58*, 458–469. [\[CrossRef\]](#)
7. Ng, C.T.; Cheng, T.C.E.; Yuan, J.J. A note on the complexity of the problem of two-agent scheduling on a single machine. *J. Comb. Optim.* **2006**, *12*, 387–394. [\[CrossRef\]](#)
8. Agnetis, A.; Chen, B.; Nicosia, G.; Pacifici, A. Price of fairness in two-agent single-machine scheduling problems. *Eur. J. Oper. Res.* **2019**, *276*, 79–87. [\[CrossRef\]](#)
9. Zhang, Y.B.; Zhang, Z.; Liu, Z.H. The price of fairness for a two-agent scheduling game minimizing total completion time. *J. Comb. Optim.* **2020**. [\[CrossRef\]](#)
10. Mor, B.; Mosheiov, G. Minimizing maximum cost on a single machine with two competing agents and job rejection. *J. Oper. Res. Soc.* **2016**, *67*, 1524–1531. [\[CrossRef\]](#)

11. Gao, Y.; Yuan, J.J. Bi-criteria Pareto-scheduling on a single machine with due indices and precedence constraints. *Discret. Optim.* **2017**, *25*, 105–119. [\[CrossRef\]](#)
12. Agnetis, A.; Billaut, J.; Gawiejnowicz, S.; Pacciarelli, D.; Soukhal, A. *Multiagent Scheduling-Models and Algorithms*; Springer: Berlin/Heidelberg, Germany, 2014.
13. Liu, P.; Gu, M.; Li, G.G. Two-agent scheduling on a single machine with release dates. *Comput. Oper. Res.* **2019**, *111*, 35–42. [\[CrossRef\]](#)
14. Oron, D.; Shabtay, D.; Steiner, G. Single machine scheduling with two competing agents and equal jobs processing times. *Eur. J. Oper. Res.* **2015**, *244*, 86–99. [\[CrossRef\]](#)
15. Perez-Gonzalez, P.; Framinan, J. A common framework and taxonomy for multicriteria scheduling problem with interfering and competing jobs: multi-agent scheduling problems. *Eur. J. Oper. Res.* **2014**, *235*, 1–16. [\[CrossRef\]](#)
16. Yuan, J.J. Complexities of some problems on multi-agent scheduling on a single machine. *J. Oper. Res. Soc. China* **2016**, *4*, 379–384. [\[CrossRef\]](#)
17. Yuan, J.J. Complexities of four problems on two-agent scheduling. *Optim. Lett.* **2018**, *12*, 763–780. [\[CrossRef\]](#)
18. Blazewicz, J.; Finke, G. Minimizing mean weighted execution time loss on identical and uniform processors. *Inform. Process. Lett.* **1987**, *24*, 259–263. [\[CrossRef\]](#)
19. Potts, C.N.; Van Wassenhove, L.N. Single machine scheduling to minimize total late work. *Oper. Res.* **1992**, *40*, 586–595. [\[CrossRef\]](#)
20. Potts, C.N.; Van Wassenhove, L.N. Approximation algorithms for scheduling a single machine to minimize total late work. *Oper. Res. Lett.* **1991**, *11*, 261–266. [\[CrossRef\]](#)
21. Hariri, A.M.A.; Potts, C.N.; Van Wassenhove, L.N. Single machine scheduling to minimize total weighted late work. *ORSA J. Comput.* **1995**, *7*, 232. [\[CrossRef\]](#)
22. Sterna, M. A survey of scheduling problems with late work criteria. *Omega* **2011**, *39*, 120–129. [\[CrossRef\]](#)
23. Wang, D.J.; Kang, C.C.; Shiau, Y.R.; Wu, C.C. A two-agent single-machine scheduling problem with late work criteria. *Soft Comput.* **2017**, *21*, 2015–2033. [\[CrossRef\]](#)
24. Zhang, X.G.; Wang, Y. Two-agent scheduling problems on a single-machine to minimize the total weighted late work. *J. Comb. Optim.* **2017**, *33*, 945–955.
25. Zhang, Y.; Yuan, J.J. A note on a two-agent scheduling problem related to the total weighted late work. *J. Comb. Optim.* **2019**, *37*, 989–999. [\[CrossRef\]](#)
26. Lee, C.Y. Machine scheduling with an availability constraints. *J. Glob. Optim.* **1996**, *9*, 395–416. [\[CrossRef\]](#)

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).