Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 471016, 7 pages
http://dx.doi.org/10.1155/2014/471016

Research Article

Some Single-Machine Scheduling Problems with Learning Effects and Two Competing Agents

Hongjie Li,[1] Zeyuan Li,[2] and Yunqiang Yin[3,4]

[1] *Mathematics Department, Zhoukou Normal University, Zhoukou 466001, China*
[2] *College of Mechanical and Electrical, Harbin Institute of Technology, Harbin 150001, China*
[3] *College of Sciences, East China Institute of Technology, Nanchang 330013, China*
[4] *Faculty of Science, Kunming University of Science and Technology, Kunming 650093, China*

Correspondence should be addressed to Yunqiang Yin; yunqiangyin@gmail.com

Received 29 April 2014; Accepted 10 May 2014; Published 25 May 2014

Academic Editor: Dar-Li Yang

This study considers a scheduling environment in which there are two agents and a set of jobs, each of which belongs to one of the two agents and its actual processing time is defined as a decreasing linear function of its starting time. Each of the two agents competes to process its respective jobs on a single machine and has its own scheduling objective to optimize. The objective is to assign the jobs so that the resulting schedule performs well with respect to the objectives of both agents. The objective functions addressed in this study include the maximum cost, the total weighted completion time, and the discounted total weighted completion time. We investigate three problems arising from different combinations of the objectives of the two agents. The computational complexity of the problems is discussed and solution algorithms where possible are presented.

1. Introduction

In traditional scheduling research, it is commonly assumed that the processing times of the jobs remain unchanged throughout the scheduling horizon. However, under certain circumstances, the job processing times may become short due to learning effects in the production environment. For example, Biskup [1] points out that the repeated processing of similar tasks will improve workers' efficiency; that is, it takes workers shorter times to process setups, operate machines or software, or handle raw materials and components. In such an environment, a job scheduled later will consume less time than the same job when scheduled earlier. Jobs in such a setting are said to be under the "learning effect" in the literature.

Biskup [1] and Cheng and Wang [2] first introduce the idea of learning into the field of scheduling independently. Since then, a large body of literature on scheduling with learning effects has emerged. Examples of such studies are Mosheiov [3], Mosheiov and Sidney [4], Bachman and Janiak [5], Janiak and Rudek [6], Wang [7], and Yin et al. [8]. Biskup [9] provides a comprehensive review of research on scheduling with learning effects. For more recent studies in this line of research, the reader is referred to Jiang et al. [10, 11], Yang [12], S.-J. Yang and D.-L. Yang [13], Wang et al. [14], Wu et al. [15], Xu et al. [16], and Yin et al. [8].

All the above papers consider the traditional case with a single agent. In recent years scheduling researchers have increasingly considered the setting of multiple competing agents, in which multiple agents need to process their own sets of jobs, competing for the use of a common resource and each agent has its own objective to optimize. However, there is little scheduling research in the multiagent setting in which the jobs are under learning effects. Liu et al. [17] study two models with two agents and position-dependent processing times. They assume that the actual processing time of job J_j is p_j+br in the aging-effect model, while the actual processing time of J_j is $p_j - br$ in the learning-effect model, where r represents the processed position of J_j and $b > 0$ denotes the aging or learning index. Ho et al. [18] define the actual processing time of job J_j as $p_j = a_j(1-kt)$ if it is processed at time t, where a_j denotes the normal processing time of job J_j

and $k \geq 0$ represents a constant such that $k(t_0 + \sum_{j=1}^{n} a_j - a_{\min}) < 1$ with $a_{\min} = \min_{j=1,2,\ldots,n}\{a_j\}$. Inspired by Ho et al. [18], Yin et al. [19] consider some two-agent scheduling problems under the learning effect model proposed in Ho et al. [18], in which the objective functions for agent A include the maximum earliness cost, the total earliness cost, and the total weighted earliness cost, and the objective function for agent B is always the same, that is, maximum earliness cost, and the objective is to minimize the objective function of agent A while keeping the objective function of agent B not greater than a given level. Similar models have been further studied by Wang and Xia [20], Wang [21], and so on. For the other related two-agent works without time-dependent processing times, the reader can refer to Baker and Smith [22], Agnetis et al. [23, 24], Cheng et al. [25, 26], Ng et al. [27], Mor and Mosheiov [28], Lee et al. [29], Leung et al. [30], Wan et al. [31], Yin et al. [19, 32], Yu et al. [33], and Zhao and Lu [34].

This study introduces a new scheduling model in which both the two-agent concept and the learning effects exist, simultaneously. We consider the following objective functions: the maximum cost, total completion time, total weighted completion time, and discounted total weighted completion time. The structural properties of optimal schedules are derived and polynomial time algorithms are developed for the problems where possible.

The remaining part of the study is organized as follows: Section 2 introduces the notation and terminology used throughout the paper. Sections 3–6 analyze the computational complexity and derive the optimal properties of the problems under study. The last section concludes the paper and suggests topics for future research.

2. Model Formulation

The problem investigated in this paper can be formally described as follows. Suppose that there are two agents A and B, each of whom has a set of nonpreemptive jobs. The two agents compete to process their jobs on a common machine. Agent A has to process the job set $J^A = \{J_1^A, J_2^A, \ldots, J_{n_A}^A\}$, while agent B has to process the job set $J^B = \{J_1^B, J_2^B, \ldots, J_{n_B}^B\}$. All the jobs are available for processing at time t_0, where $t_0 \geq 0$. Let $X \in \{A, B\}$. The jobs belonging to agent X are called A-jobs. Associated with each job J_j^X ($j \in \{1, 2, \ldots, n_X\}$), there are normal processing time a_j^X and weight w_j^X. Due to the learning effect, the actual processing time p_j of job J_j^X is defined as

$$p_j^X = a_j^X (1 - kt), \quad j = 1, 2, \ldots, n_X, \tag{1}$$

where $t \geq t_0$ denotes job's starting time and $k \geq 0$ represents constant such that $k(t_0 + \sum_{J_j^X \in J^A \cup J^B} a_j^X - a_{\min}) < 1$, where $a_{\min} = \min_{J_j^X \in J^A \cup J^B}\{a_j^X\}$ (see Ho et al. [18] for details).

Given a feasible schedule S of the $n = n_A + n_B$ jobs, we use $C_j^X(S)$ to denote the completion time of job J_j^X and omit the argument S whenever this does not cause confusion. The makespan of agent X is $C_{\max}^X = \max_{j=1,2,\ldots,n_X}\{C_j^X\}$. For each job J_j^X, let $f_j^X(\cdot)$ be a nondecreasing function. In this case, the maximum cost is defined as $f_{\max}^X = \max_{j=1,2,\ldots,n_X}\{f_j^X(C_j^X)\}$. The objective function of agent X considered in this paper includes the following: $f_{\max}^X$ (maximum cost), $\sum C_j^X$ (total completion time), $\sum w_j^X C_j^X$ (total weighted completion time), and $\sum w_j^X (1 - e^{-rC_j^X})$ (discounted total weighted completion time).

Using the three-field notation scheme $\alpha|\beta|\gamma$ introduced by Graham et al. [35], the problems considered in this paper are denoted as follows: $1 \mid p_j^X = a_j^X(1 - kt) \mid f_{\max}^A(C^A) : f_{\max}^B(C^B) \leq U$, $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum C_j^A : f_{\max}^B(C^B) \leq U$, $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum w_j^A C_j^A : f_{\max}^B(C^B) \leq U$, and $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum w_j^A (1 - e^{-rC_j^A}) : f_{\max}^B(C^B) \leq U$.

Note that all the objective functions involved in the considered problems are regular; that is, they are nondecreasing in the job completion times. Hence there is no benefit in keeping the machine idle.

3. Problem $1 \mid p_j^X = a_j^X(1-kt) \mid f_{\max}^A : f_{\max}^B \leq U$

In this section we address the problem $1 \mid p_j^X = a_j^X(1 - kt) \mid f_{\max}^A : f_{\max}^B \leq U$ and show that it can be solved optimally in polynomial time. We first develop some structural properties of optimal schedules for the problem which will be used in developing the algorithm.

Lemma 1 (see [19]). *For problem $1 \mid p_j^X = a_j^X(1 - kt) \mid C_{\max}$, the makespan is equal to*

$$\begin{aligned}&\left(t_0 - \frac{1}{k}\right)\prod_{j=1}^{n_X}\left(1 - ka_j^X\right) + \frac{1}{k}\\ &\quad = \left(t_0 - \frac{1}{k}\right)\prod_{j=1}^{n_A}\left(1 - ka_j^A\right)\prod_{j=1}^{n_B}\left(1 - ka_j^B\right) + \frac{1}{k}.\end{aligned} \tag{2}$$

In the sequel, we set $u = (t_0 - (1/k))\prod_{j=1}^{n_X}(1 - ka_j^X) + (1/k)$. Then the following results hold.

Proposition 2. *For the problem $1 \mid p_j^X = a_j^X(1 - kt) \mid f_{\max}^A : f_{\max}^B \leq U$, if there is a B-job J_k^B such that $f_k^B(u) \leq U$, then there exists an optimal schedule such that J_k^B is scheduled last and there is no optimal schedule where an A-job is scheduled last.*

Proof. Assume that S is an optimal schedule where the B-job J_h^B is not scheduled in the last position. Let π denote the set of jobs scheduled prior to job J_h^B. We construct from S a new schedule S' by moving job J_h^B to the last position and leaving the other jobs unchanged in S. Then, the completion times of the jobs processed before job J_h^B in S' are the same as that in S since there is no change for any job preceding J_h^B in S. The jobs belonging to π are scheduled earlier, so their completion times are smaller in S' by Lemma 1. It follows that $f_k^X(C_k^X(S')) \leq f_k^X(C_k^X(S))$ for any job J_k^X in π, where $X \in \{A, B\}$. By the assumption that $f_h^B(u) \leq U$, job J_h^B

is feasible in S', so schedule S' is feasible and optimal, as required. □

For each B-job J_j^B, let us define a deadline D_j^B such that $f_j^B(C_j^B) \le U$ for $C_j^B \le D_j^B$ and $f_j^B(C_j^B) > U$ for $C_j^B > D_j^B$ (if the inverse function $f_j^B(\cdot)$ is available, the deadlines can be evaluated in constant time; otherwise, this requires logarithmic time).

Proposition 3. *For the problem* $1 \mid p_j^X = a_j^X(1-kt) \mid f_{\max}^A : f_{\max}^B \le U$, *there exists an optimal schedule where the B-jobs are scheduled according to the nondecreasing order of* D_j^B.

Proof. Assume that S is an optimal schedule where the B-jobs are not scheduled according to the nondecreasing order of D_j^B. Let J_l^B and J_h^B be the first pair of jobs such that $D_l^B > D_h^B$. In this schedule, job J_l^B is processed earlier; then a set of A-jobs, denoted as π, are consecutively processed and then job J_h^B. In addition, denote by π' the set of jobs processed after job J_h^B. We construct from S a new schedule S' by extracting job J_l^B, reinserting it just after job J_h^B and leaving the other jobs unchanged in schedule S. Then the completion times of the jobs processed prior to job J_l^B in S' are the same as that in S. By Lemma 1, the completion time of job J_h^B in S equals that of job J_l^B in S'; that is, $C_l^B(S') = C_h^B(S)$, so the completion times of the jobs belonging to π' are identical in both S and S'. Since S is feasible, it follows that $C_l^B(S') = C_h^B(S) \le D_h^B < D_l^B$, so job J_l^B is feasible in S'. The π-jobs and job J_h^B are scheduled earlier in S', implying that their actual processing times are smaller in S', so their completion times are earlier in S', and thus they remain feasible. Therefore, schedule S' is feasible and optimal.

Thus, repeating doing this procedure for all the B-jobs not sequenced according to nondecreasing order of D_j^B completes the proof. □

Proposition 4. *For the problem* $1 \mid p_j^X = a_j^X(1-kt) \mid f_{\max}^A : f_{\max}^B \le U$, *if* $f_k^B(u) > U$ *for any B-job* J_k^B, *then there exists an optimal schedule where the A-job with the smallest cost* $f_k^A(u)$ *is processed in the last position.*

Proof. Assume that S is an optimal schedule where the A-job with the smallest cost J_h^A, that is, $f_h^A(u) = \min_{J_j^A \in J^A}\{f_j^A(u)\}$, is not processed in the last position. By the hypothesis, the last job in schedule S is an A-job, say J_l^A. This means $f_h^A(u) < f_l^A(u)$. In this schedule, job J_h^A is scheduled earlier. Let π denote the set of jobs scheduled after job J_h^A and prior to job J_l^A. We construct from S a new schedule S' by extracting job J_h^A, reinserting it just after job J_l^A, and leaving the other jobs unchanged in schedule S. There is no change for any job preceding J_h^A in S. We claim the following.

(1) Schedule S' is feasible. First, the completion times of the jobs processed prior to job J_h^A in S' are the same as that in S'. Since the jobs belonging to π are scheduled earlier in S', their actual processing times are smaller in S', so their completion times are earlier in S'. It follows that $f_k^X(C_k^X(S')) \le f_k^X(C_k^X(S))$ for any job J_k^X in π, where $X \in \{A, B\}$, as required.

(2) Schedule S' is a better schedule than S. By Lemma 1, the completion time of the last job J_l^A in S equals that of the last job J_h^A in S'; that is, $C_l^A(S) = C_h^A(S') = u$. Thus, to prove that S' is better than S, it suffices to show that

$$\begin{aligned} &\max\left\{f_l^A\left(C_l^X\left(S'\right)\right), f_h^A(u)\right\} \\ &\quad \le \max\left\{f_h^A\left(C_h^X\left(S'\right)\right), f_l^A(u)\right\}. \end{aligned} \tag{3}$$

Since $f_k^A(\cdot)$ is a nondecreasing function of the completion time of job J_k^A and $C_l^X(S') < u$, we have $f_l^A(C_l^X(S')) \le f_l^A(u)$. Thus $\max\{f_l^A(C_l^X(S')), f_h^A(u)\} \le f_l^A(u)$, as required.

The result follows. □

Summing up the above analysis, our algorithm for problem $1 \mid p_j^X = a_j^X(1-kt) \mid f_{\max}^A : f_{\max}^B \le U$ can be formally described as in Algorithm 1.

Theorem 5. *Algorithm 1 solves problem* $1 \mid p_j^X = a_j^X(1-kt) \mid f_{\max}^A : f_{\max}^B \le U$ *in* $O(n_A^2 + n_B \log n_B)$ *time.*

Proof. Step 1 requires a sorting operation of the B-jobs, which takes $O(n_B \log n_B)$ time. Step 2 takes $O(n_B)$ time since the calculation of the $f_j^B(\cdot)$ functions in Step 2 can be evaluated in constant time by the assumption. In Step 3 we calculate the $f_j^A(\cdot)$ value for all the remaining unscheduled A-jobs, which takes $O(n_A)$ time. Thus, after n_A iterations, Step 3 can be executed in $O(n_A^2)$ time. Therefore, the overall time complexity of the algorithm is indeed $O(n_A^2 + n_B \log n_B)$. □

4. Problem $1 \mid p_j^X = a_j^X(1-kt) \mid \sum w_j^A C_j^A : f_{\max}^B \le U$

Leung et al. [30] show that problem $1 \| \sum w_j^A C_j^A : f_{\max}^B \le U$ is NP-hard in the strong sense. Since our problem $1 \mid p_j^X = a_j^X(1-kt) \mid \sum w_j^A C_j^A : f_{\max}^B \le U$ is a generalization of the problem $1 \| \sum w_j^A C_j^A : f_{\max}^B \le U$, then so is our problem. In what follows we show that the problem $1 \mid p_j^X = a_j^X(1-kt) \mid \sum w_j^A C_j^A : f_{\max}^B \le U$ is polynomially solvable if the A-jobs have reversely agreeable weights; that is, $a_i^A \le b_j^A$ implies $w_i^A \ge w_j^A$ for all jobs J_i^A and J_j^A. It is clear that Propositions 2 and 3 still hold for this problem. We modify Proposition 4 as follows.

Proposition 6. *For the problem* $1 \mid p_j^X = a_j^X(1-kt) \mid \sum w_j^A C_j^A : f_{\max}^B \le U$, *if the A-jobs have reversely agreeable weights, then there exists an optimal schedule where the A-jobs are assigned according to the nondecreasing order of* a_j^A/w_j^A, *that is, in the weighted shortest processing time (WSPT) order.*

Input: $n_A, n_B, U, p^A = (p_1^A, p_2^A, \ldots, p_{n_A}^A)$ and $p^B = (p_1^B, p_2^B, \ldots, p_{n_B}^B)$.
Step 1. Set $h = n_B$, $J = J^A$, $f_{\max}^A = 0$ and $t = (t_0 - 1/k)\prod_{j=1}^{n_X}(1 - ka_j^X) + 1/k$,; solve D_j^B from $f_j^B(D_j^B) = U$ for $j = 1, 2, \ldots, n_B$ and renumber them according to the non-decreasing order such that $D_{[1]}^B \le D_{[2]}^B \le \cdots \le D_{[n_B]}^B$;
Step 2. If $h \ge 1$, then
  If $t \le D_{[h]}^B$, then
    set $h = h - 1$, $t = (t - a_{[h]}^B)/(1 - ka_{[h]}^B)$, assign job $J_{[h]}^B$ at time t, and go to Step 2;
  Else
    go to Step 3;
Else
  go to Step 3;
Step 3. If $J \ne \emptyset$, then
  select the job J_l^A from J with the the smallest cost, that is, $f_l^A(t) = \min_{J_j^A \in J^A}\{f_j^A(t)\}$,
  set $f_{\max}^A = \max\{f_{\max}^A, f_l^A(t)\}$, $t = (t - a_l^A)/(1 - ka_l^A)$, assign job J_l^A at time t, delete J_l^A from J, and go to Step 4;
Elseif $h \ge 1$
  report that the instance is not feasible;
Else
  go to Step 4;
Step 4. If J is not empty or $h \ge 1$, then
  go to Step 2.
Else
  stop.

Algorithm 1

Proof. Assume that S is an optimal schedule where A-jobs are not scheduled in the WSPT order. Let J_l^A and J_h^A be the first pair of jobs such that $a_l^A/w_l^A > a_h^A/w_h^A$. Then $a_l^A \ge a_h^A$ and $w_l^A \le w_h^A$ due to the fact that the A-jobs have reversely agreeable weights. Assume that, in schedule S, job J_l^A starts its processing at time T; then a set of B-jobs are consecutively processed and then job J_h^A. In addition, let π' denote the set of jobs processed after job J_h^A. We construct a new scheduling S' from S by swapping jobs J_l^A and J_h^A and leaving the other jobs unchanged. We conclude the following.

(1) Schedule S' is feasible. By Lemma 1, the completion time of job J_h^A in S equals that of job J_l^A in S'; that is, $C_l^A(S') = C_h^A(S)$, so the completion times of the jobs belonging to π' are identical in both S and S'. Since $a_l^A \ge a_h^A$, we have $C_h^A(S') = T + a_h^A(1 - kT) \le T + a_l^A(1 - kT) = C_l^A(S)$. Hence the π-jobs are scheduled earlier in S', implying that their actual processing times are smaller in S', so their completion times are earlier in S'. Hence $f_k^B(C_k^B(S')) \le f_k^B(C_k^B(S))$ for any job J_k^B in π, as required.

(2) Schedule S' is better than S. By the proof of (1), it is sufficient to show that

$$w_h^A C_h^A(S') + w_l^A C_l^A(S') \le w_l^A C_l^A(S) + w_h^A C_h^A(S). \quad (4)$$

Since $C_h^A(S') \le C_l^A(S)$ and $C_l^A(S') = C_h^A(S)$, we have

$$\begin{aligned} &w_l^A C_l^A(S) + w_h^A C_h^A(S) - \left(w_h^A C_h^A(S') + w_l^A C_l^A(S')\right) \\ &\quad \ge w_l^A C_h^A(S') + w_h^A C_l^A(S') - \left(w_h^A C_h^A(S') + w_l^A C_l^A(S')\right) \\ &\quad = \left(w_l^A - w_h^A\right)\left(C_h^A(S') - C_l^A(S')\right) \\ &\quad \ge 0, \end{aligned} \quad (5)$$

as required.

Thus, repeating this swapping argument for all the A-jobs not sequenced in the WSPT order yields the theorem. □

Based on the results of Propositions 2, 3, and 6, our algorithm to solve the problem $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum w_j^A C_j^A : f_{\max}^B \le U$ for the case where the A-jobs have reversely agreeable weights can be formally described as in Algorithm 2.

Theorem 7. *The problem* $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum w_j C_j^A : f_{\max}^B \le U$ *can be solved in* $O(n_A \log n_A + n_B \log n_B)$ *time by applying Algorithm 2 if all A-jobs have reversely agreeable weights.*

Proof. The correctness comes from the above analysis. Now we turn to the time complexity of the algorithm. Step 1 requires two sorting operations of the A-jobs and B-jobs, respectively, which take $O(n_A \log n_A)$ time and $O(n_B \log n_B)$ time, respectively. Both Steps 2 and 3 take $O(2)$ time.

Input: $n_A, n_B, U, w^A = (w_1^A, w_2^A, \ldots, w_{n_A}^A)$, $p^A = (p_1^A, p_2^A, \ldots, p_{n_A}^A)$ and $p^B = (p_1^B, p_2^B, \ldots, p_{n_B}^B)$.
Step 1. Set $t = (t_0 - 1/k)\prod_{j=1}^{n_X}(1 - ka_j^X) + 1/k, l = n_A, h = n_B$, and $\sum w_j^A C_j^A = 0$;
sort the A-jobs according to the non-decreasing order of a_j^A/w_j^A, that is,
$a_{[1]}^A/w_{[1]}^A \le a_{[2]}^A/w_{[2]}^A \le \cdots \le a_{[n_A]}^A/w_{[n_A]}^A$;
calculate the deadlines of the B-jobs from $f_j^B(D_j^B) = U$ and renumber them
according to the non-decreasing order such that $D_{[1]}^B \le D_{[2]}^B \le \cdots \le D_{[n_B]}^B$;
Step 2. If $h \ge 1$, then
If $t \le D_{[h]}^B$, then
set $h = h - 1$, $t = (t - a_{[h]}^B)/(1 - ka_{[h]}^B)$, assign job $J_{[h]}^B$ at time t, and go to Step 2;
Else
go to Step 3;
Else
go to Step 3;
Step 3. If $l \ge 1$, then
set $l = l - 1$, $\sum w_j^A C_j^A = \sum w_j^A C_j^A + w_{[l]}^A t$, $t = (t - a_{[l]}^A)/(1 - ka_{[l]}^A)$, assign $J_{[l]}^A$ at time t, and go to Step 4;
Elseif $h \ge 1$
output that the instance is not feasible;
Else
go to Step 4;
Step 4. If $h \ge 1$ or $l \ge 1$, then
go to Step 2.
Else
stop.

Algorithm 2

Therefore, the overall time complexity of the algorithm is indeed $O(n_A \log n_A + n_B \log n_B)$. □

5. Problem $1 \mid p_j^X = a_j^X(1-kt) \mid \sum w_j^A(1-e^{-rC_j^A}) : f_{\max}^B \le U$

This section address the problem $1 \mid p_j^X = a_j^X(1 - kt) \mid w_j^A(1 - e^{-rC_j^A}) : f_{\max}^B \le U$. We show that it is polynomially solvable if the A-jobs have reversely agreeable weights. It is clear that Propositions 2 and 3 still hold for this problem. We give Proposition 8 as follows.

Proposition 8. *For the problem $1 \mid p_j^X = a_j^X(1 - kt) \mid w_j^A(1 - e^{-rC_j^A}) : f_{\max}^B \le U$, if the A-jobs have reversely agreeable weights, then there exists an optimal schedule where the A-jobs are assigned according to the nondecreasing order of $(1 - e^{-ra_j^A})/w_j^A e^{-ra_j^A}$, that is, in the weighted discount shortest processing time (WDSPT) order.*

Proof. We adopt the same notation as that used in the proof of Proposition 6. Assume that $(1 - e^{-ra_l^A})/w_l^A e^{-ra_l^A} > (1 - e^{-rp_h^A})/w_h^A e^{-rp_h^A}$. Since A-jobs have reversely agreeable weights, we have $a_l^A \ge a_h^A$ and $w_l^A \le w_h^A$. Then by the proof of Proposition 6, we know that $C_h^A(S') \le C_l^A(S)$, $C_l^A(S') = C_h^A(S)$, and $C_k^A(S') = C_j^A(S)$ for all the other jobs $J_k^A \in J_A/\{J_l^A, J_k^A\}$ and that schedule S' is feasible. To show that S' is better than S, it is sufficient to show that

$$\begin{aligned} &w_h^A\left(1 - e^{-rC_h^A(S')}\right) + w_l^A\left(1 - e^{-rC_l^A(S')}\right) \\ &\quad \le w_l^A\left(1 - e^{-rC_l^A(S)}\right) + w_h^A\left(1 - e^{-rC_h^A(S)}\right). \end{aligned} \tag{6}$$

In fact, since $r \in (0, 1)$, $C_h^A(S') \le C_l^A(S)$, and $C_l^A(S') = C_h^A(S)$, we have

$$\begin{aligned} &w_l^A\left(1 - e^{-rC_l^A(S)}\right) + w_h^A\left(1 - e^{-rC_h^A(S)}\right) \\ &\quad - \left(w_h^A\left(1 - e^{-rC_h^A(S')}\right) + w_l^A\left(1 - e^{-rC_l^A(S')}\right)\right) \\ &= w_h^A e^{-rC_h^A(S')} + w_l^A e^{-rC_l^A(S')} - w_l^A e^{-rC_l^A(S)} - w_h^A e^{-rC_h^A(S)} \\ &\ge w_h^A e^{-rC_h^A(S')} + w_l^A e^{-rC_h^A(S')} - w_l^A e^{-rC_h^A(S')} - w_h^A e^{-rC_l^A(S')} \\ &= \left(w_h^A - w_l^A\right)\left(e^{-rC_h^A(S')} - e^{-rC_l^A(S')}\right) \\ &\ge 0. \end{aligned} \tag{7}$$

Hence, $w_h^A(1 - e^{-rC_h^A(S')}) + w_l^A(1 - e^{-rC_l^A(S')}) \le w_l^A(1 - e^{-rC_l^A(S)}) + w_h^A(1 - e^{-rC_h^A(S)})$. Therefore, S' is not worse than S. Thus, repeating this swapping argument for all the A-jobs not sequenced in the WDSPT order yields the theorem. □

Based on the above analysis, our algorithm to solve the problem $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum w_j^A(1 - e^{-rC_j^A}) : f_{\max}^B \le U$ for the case where the A-jobs have reversely agreeable weights can be described as in Algorithm 3.

Input: $n_A, n_B, U, r, w^A = (w_1^A, w_2^A, \ldots, w_{n_A}^A)$, $p^A = (p_1^A, p_2^A, \ldots, p_{n_A}^A)$ and $p^B = (p_1^B, p_2^B, \ldots, p_{n_B}^B)$.

Step 1. Set $t = (t_0 - 1/k)\prod_{j=1}^{n_X}(1 - ka_j^X) + 1/k, l = n_A, h = n_B$, and $\sum w_j^A(1 - e^{-rC_j^A}) = 0$;
sort the A-jobs in non-decreasing order of $(1 - e^{-ra_j^A})/(w_j^A e^{-ra_j^A})$,
that is, $(1 - e^{-ra_{[1]}^A})/(w_{[1]}^A e^{-ra_{[1]}^A}) \le (1 - e^{-ra_{[2]}^A})/(w_{[2]}^A e^{-ra_{[2]}^A}) \le \cdots \le (1 - e^{-ra_{[n_A]}^A})/(w_{[n_A]}^A e^{-ra_{[n_A]}^A})$;
calculate the deadlines of the B-jobs from $f_j^B(D_j^B) = U$ and
renumber them according to the non-decreasing order, that is, $D_{[1]}^B \le D_{[2]}^B \le \cdots \le D_{[n_B]}^B$;

Step 2. If $h \ge 1$, then
  If $t \le D_{[h]}^B$, then
    set $h = h - 1$, $t = (t - a_{[h]}^B)/(1 - ka_{[h]}^B)$, assign $J_{[h]}^B$ at time t, and go to Step 2;
  Else
    go to Step 3;
Else
  go to Step 3;

Step 3. If $l \ge 1$, then
  set $l = l - 1$, $\sum w_j^A(1 - e^{-rC_j^A}) = \sum w_j^A(1 - e^{-rC_j^A}) + w_{[l]}^A(1 - e^{-rt})$, $t = (t - a_{[l]}^A)/(1 - ka_{[l]}^A)$,
  assign $J_{[l]}^A$ at time t, and go to Step 4;
Elseif $h \ge 1$
  output that the instance is not feasible;
Else
  go to Step 4;

Step 4. If $h \ge 1$ or $l \ge 1$, then
  go to Step 2.
Else
  stop.

Algorithm 3

Theorem 9. *The problem* $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum w_j^A(1 - e^{-rC_j^A}) : f_{\max}^B \le U$ *can be solved in* $O(n_A \log n_A + n_B \log n_B)$ *time by applying Algorithm 3 for the case that the A-jobs have reversely agreeable weights.*

Proof. The proof is analogous to that of Theorem 7. □

6. Conclusions

This paper introduced a new scheduling model on a single machine that involves two agents and learning effects simultaneously. We studied the problem of finding an optimal schedule for agent A, subject to the constraint that the maximum cost of agent B does not exceed a given value. We derived the optimal structural properties of optimal schedules and provided polynomial time algorithms for the problem $1 \mid p_j^X = a_j^X(1 - kt) \mid f_{\max}^A : f_{\max}^B \le U$. We also showed that the problems $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum w_j C_j^A : f_{\max}^B \le U$ and $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum w_j^A(1 - e^{-rC_j^A}) : f_{\max}^B \le U$ can also be solved in polynomial time under certain agreeable conditions. Future research may consider the scheduling model with more than two agents.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are grateful to the Editor and two anonymous referees for their helpful comments on an earlier version of their paper. This research was supported by the National Natural Science Foundation of China (no. 71301022).

References

[1] D. Biskup, "Single-machine scheduling with learning considerations," *European Journal of Operational Research*, vol. 115, no. 1, pp. 173–178, 1999.

[2] T. C. E. Cheng and G. Wang, "Single machine scheduling with learning effect considerations," *Annals of Operations Research*, vol. 98, no. 1–4, pp. 273–290, 2000.

[3] G. Mosheiov, "Scheduling problems with a learning effect," *European Journal of Operational Research*, vol. 132, no. 3, pp. 687–693, 2001.

[4] G. Mosheiov and J. B. Sidney, "Scheduling with general job-dependent learning curves," *European Journal of Operational Research*, vol. 147, no. 3, pp. 665–670, 2003.

[5] A. Bachman and A. Janiak, "Scheduling jobs with position-dependentprocessing times," *Journal of the Operational Research Society*, vol. 55, no. 3, pp. 257–264, 2004.

[6] A. Janiak and R. Rudek, "A new approach to the learning effect: beyond the learning curve restrictions," *Computers and Operations Research*, vol. 35, no. 11, pp. 3727–3736, 2008.

[7] J.-B. Wang, "Single-machine scheduling with general learning functions," *Computers and Mathematics with Applications*, vol. 56, no. 8, pp. 1941–1947, 2008.

[8] Y. Yin, D. Xu, K. Sun, and H. Li, "Some scheduling problems with general position-dependent and time-dependent learning effects," *Information Sciences*, vol. 179, no. 14, pp. 2416–2425, 2009.

[9] D. Biskup, "A state-of-the-art review on scheduling with learning effects," *European Journal of Operational Research*, vol. 188, no. 2, pp. 315–329, 2008.

[10] Z. Jiang, F. Chen, and H. Kang, "Single-machine scheduling problems with actual time-dependent and job-dependent learning effect," *European Journal of Operational Research*, vol. 227, no. 1, pp. 76–80, 2013.

[11] Z. Jiang, F. Chen, and C. Wu, "Minimizing the maximum lateness in a single-machine scheduling problem with the normal time-dependent and job-dependent learning effect," *Applied Mathematics and Computation*, vol. 218, no. 18, pp. 9438–9441, 2012.

[12] S.-J. Yang, "Single-machine scheduling problems with both start-time dependent learning and position dependent aging effects under deteriorating maintenance consideration," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3321–3329, 2010.

[13] S.-J. Yang and D.-L. Yang, "Minimizing the makespan on single-machine scheduling with aging effect and variable maintenance activities," *Omega*, vol. 38, no. 6, pp. 528–533, 2010.

[14] J.-B. Wang, L. Sun, and L. Sun, "Single machine scheduling with exponential sum-of-logarithm-processing-times based learning effect," *Applied Mathematical Modelling*, vol. 34, no. 10, pp. 2813–2819, 2010.

[15] C.-C. Wu, Y. Yin, and S.-R. Cheng, "Single-machine and two-machine flowshop scheduling problems with truncated position-based learning functions," *Journal of the Operational Research Society*, vol. 64, no. 1, pp. 147–156, 2013.

[16] D. Xu, Y. Yin, and H. Li, "Comments on "a note on minimizing maximum lateness in an m-machine scheduling problem with a learning effect"," *Applied Mathematics and Computation*, vol. 217, no. 2, pp. 939–943, 2010.

[17] P. Liu, X. Zhou, and L. Tang, "Two-agent single-machine scheduling with position-dependent processing times," *International Journal of Advanced Manufacturing Technology*, vol. 48, no. 1–4, pp. 325–331, 2010.

[18] K. I.-J. Ho, J. Y.-T. Leung, and W.-D. Wei, "Complexity of scheduling tasks with time-dependent execution times," *Information Processing Letters*, vol. 48, no. 6, pp. 315–320, 1993.

[19] Y. Yin, S.-R. Cheng, and C.-C. Wu, "Scheduling problems with two agents and a linear non-increasing deterioration to minimize earliness penalties," *Information Sciences*, vol. 189, pp. 282–292, 2012.

[20] J.-B. Wang and Z.-Q. Xia, "Scheduling jobs under decreasing linear deterioration," *Information Processing Letters*, vol. 94, no. 2, pp. 63–69, 2005.

[21] J.-B. Wang, "A note on single-machine scheduling with decreasing time-dependent job processing times," *Applied Mathematical Modelling*, vol. 34, no. 2, pp. 294–300, 2010.

[22] K. R. Baker and J. C. Smith, "A multiple-criterion model for machine scheduling," *Journal of Scheduling*, vol. 6, no. 1, pp. 7–16, 2003.

[23] A. Agnetis, P. B. Mirchandani, D. Pacciarelli, and A. Pacifici, "Scheduling problems with two competing agents," *Operations Research*, vol. 52, no. 2, pp. 229–242, 2004.

[24] A. Agnetis, D. Pacciarelli, and A. Pacifici, "Multi-agent single machine scheduling," *Annals of Operations Research*, vol. 150, no. 1, pp. 3–15, 2007.

[25] T. C. E. Cheng, C. T. Ng, and J. J. Yuan, "Multi-agent scheduling on a single machine to minimize total weighted number of tardy jobs," *Theoretical Computer Science*, vol. 362, no. 1–3, pp. 273–281, 2006.

[26] T. C. E. Cheng, C. T. Ng, and J. J. Yuan, "Multi-agent scheduling on a single machine with max-form criteria," *European Journal of Operational Research*, vol. 188, no. 2, pp. 603–609, 2008.

[27] C. T. Ng, T. C. E. Cheng, and J. J. Yuan, "A note on the complexity of the problem of two-agent scheduling on a single machine," *Journal of Combinatorial Optimization*, vol. 12, no. 4, pp. 386–393, 2006.

[28] B. Mor and G. Mosheiov, "Scheduling problems with two competing agents to minimize minmax and minsum earliness measures," *European Journal of Operational Research*, vol. 206, no. 3, pp. 540–546, 2010.

[29] K. Lee, B.-C. Choi, J. Y.-T. Leung, and M. L. Pinedo, "Approximation algorithms for multi-agent scheduling to minimize total weighted completion time," *Information Processing Letters*, vol. 109, no. 16, pp. 913–917, 2009.

[30] J. Y.-T. Leung, M. Pinedo, and G. Wan, "Competitive two-agent scheduling and its applications," *Operations Research*, vol. 58, no. 2, pp. 458–469, 2010.

[31] G. Wan, S. R. Vakati, J. Y.-T. Leung, and M. Pinedo, "Scheduling two agents with controllable processing times," *European Journal of Operational Research*, vol. 205, no. 3, pp. 528–539, 2010.

[32] Y. Yin, S.-R. Cheng, T. C. E. Cheng, W.-H. Wu, and C.-C. Wu, "Two-agent single-machine scheduling with release times and deadlines," *International Journal of Shipping and Transport Logistics*, vol. 5, no. 1, pp. 75–94, 2013.

[33] X. Yu, Y. Zhang, D. Xu, and Y. Yin, "Single machine scheduling problem with two synergetic agents and piece-rate maintenance," *Applied Mathematical Modelling*, vol. 37, no. 3, pp. 1390–1399, 2013.

[34] K. Zhao and X. Lu, "Approximation schemes for two-agent scheduling on parallel machines," *Theoretical Computer Science*, vol. 468, pp. 114–121, 2013.

[35] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan, "Optimization and heuristic in deterministic sequencing and scheduling: a survey," *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.