Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 389120, 9 pages
http://dx.doi.org/10.1155/2013/389120

Research Article

Single-Machine Scheduling with Job Rejection, Deteriorating Effects, and Deteriorating Maintenance Activities

Haiwei Nian and Zhizhong Mao

School of Information Science and Engineering, Northeast University, Shenyang 110004, China

Correspondence should be addressed to Haiwei Nian; nianhw-sea@163.com

Received 19 August 2013; Accepted 6 September 2013

Academic Editor: Yunqiang Yin

This paper addresses the single-machine scheduling problems with simultaneous considerations of job rejection, deterioration effects, and deteriorating multimaintenance activities. A job is either rejected, in which case a rejection penalty has to be paid, or accepted and processed on the single machine. Three deterioration effect models are investigated, and it is assumed that each machine may be subject to several maintenance activities over the scheduling horizon, and the duration of the maintenance depends on its running time. Moreover, due to the restriction of the budget of maintenance, the upper bound of the total maintenance frequencies on the machine is assumed to be known in advance. The objective is to find jointly the optimal accepted job set, the optimal maintenance frequencies, the optimal maintenance positions, and the optimal accepted job sequence such that the cost function based on the total completion time and rejection penalty is minimized. It is shown that all the versions of the problem under study are polynomial time solutions.

1. Introduction

In classical deterministic scheduling models, the processing conditions, including the job processing times, are usually viewed as given constants. However, in many real-life situations, the processing conditions may vary over time, thereby affecting the actual durations of job processing. This leads to the study of scheduling models with variable job processing times. There are several categories of models that address scheduling problems with variable processing times. For example, the actual processing time of a job gets shorter if the job is scheduled later (such a phenomenon is called the "learning effect"). In scheduling with deterioration, the later a job starts, the longer it takes to process the job (such a job is called a "deteriorating job"). Furthermore, the job processing time can be modelled as a mixed form of both learning and deterioration. Applications of scheduling models with variable job processing times can be found in the forging process in steel plants, in silverware production, finance management, or workforce learning. The corresponding scheduling problems have received considerable research attention. For reviews of scheduling problems involving the learning effect, the reader may refer to Cheng and Wang [1], Cheng et al. [2, 3], Janiak and Rudek [4–6], Kuo et al. [7], Lee and Lai [8], Rudek [9], and Yin et al. [10, 11], while for reviews of research on scheduling with job deterioration, the reader may refer to Bachman et al. [12, 13], Inderfurth et al. [14], Janiak and Kovalyov [15], Mazdeh et al. [16], Yin et al. [17–19], and Zhao and Tang [20]. For studies considering a mixed form of both learning and deterioration, we refer the reader to Cheng et al. [21], Wang [22, 23], Wang and Cheng [24], Wang et al. [25–27], Yang and Kuo [28], and Yin and Xu [29]. The reader may also make reference to Alidaee and Womer [30], Biskup [31], and Cheng et al. [32] for recent state-of-the-art reviews in this area, as well as for discussion of practical applications of the models.

Most literature in scheduling assumes that machines are available at all times. However, machines subject to maintenance are found prevalently in process industries and manufacturing systems. In the real world, machines are usually not continuously available due to maintenance activities, tool changes, or breakdowns. It is well-known that the maintenance activity is important to improve the production efficiency of the machines or the quality of the products. During the maintenance activity, the machine is unavailable

for processing jobs. Scheduling in such an environment is specified as scheduling with availability constraints and it has attracted many researchers. For details on this stream of research, the reader may refer to the comprehensive surveys by Schmidt [33] and Ma et al. [34]. Moreover, in order to model a more realistic production system, the problem of joint scheduling with the deterioration effect and maintenance activity has received the attention of many researchers, including Cheng et al. [21], Gawiejnowicz [35], Ji et al. [36], Lee and Wu [37], Low et al. [38], Lodree and Geiger [39], Wu and Lee [40], Rustogi and Strusevich [41], S. Yang and D. Yang [42–44], and Zhao and Tang [45].

On the other hand, scheduling problems have been extensively studied in the literature under the assumption that all jobs have to be processed. However, in many cases the scheduler has to make a higher level decision on how to partition the set of jobs into a set of accepted and a set of rejected jobs and to efficiently schedule the set of accepted jobs among the machines. Job rejection may be considered, for example, in cases where the machine capacity is too low and does not allow the scheduler to process all the jobs. In such cases, the rejected jobs may be either outsourced or not get served at all, thus incurring a rejection penalty either due to an outsourcing cost or due to loss of income and reputation (see [46]). The idea of scheduling with rejection is relatively new. It was first introduced by Bartal et al. [47], who studied the problem of minimizing the sum of makespan and rejection penalties on a set of identical parallel machines, focusing on approximations. Since then, scheduling problems with rejection jobs have been extensively studied in the literature. Recently, Shabtay et al. [46] offer a very robust approach to scheduling with rejection by providing a bicriteria analysis of a large set of scheduling problems that can all be represented by (or reduced to) the same mathematical formulation.

In this problem, we investigate the scheduling problems with simultaneous considerations of rejected jobs, deterioration effects, and deteriorating multimaintenance activities which extends the problem considered in S. Yang and D. Yang [42] by allowing rejected jobs. The objective is to find jointly the accepted job set, the optimal maintenance frequencies, the optimal maintenance positions, and the optimal accepted job sequences such that the cost function based on the total completion time and rejection penalty is minimized. To the best of our knowledge, there are no papers in which such a problem has been studied so far. The rest of the paper is organized as follows: Section 2 formulates the problems under study. Section 3 addresses the optimal solution of the case where no maintenance is scheduled in the sequence. Section 4 provides polynomial time solutions for all the problems studied. We conclude the paper and suggest some future research topics in the last section.

2. Problem Definition

The single-machine scheduling with job rejection, deteriorating effects, and deteriorating maintenance activities may be stated as follows. Assume that there is a set of n independent jobs $N = \{J_1, J_2, \ldots, J_n\}$ to be processed on a single machine. The machine can handle at most one job at a time and job preemption is not allowed. Each job J_j becomes available for processing at time zero, requires a nonnegative normal processing time p_j, and has a nonnegative rejection penalty e_j. For each job we must decide either to schedule that job on the machine or to reject it. If we reject job J_j, we pay its rejection penalty e_j. We denote the sets of accepted jobs and rejected jobs by A and $\overline{A}$, respectively. Due to the deteriorating effects, maintenance may be performed on the machine to improve its production efficiency. In order to model a realistic manufacturing system, the machine may be subject to several maintenance activities during the planning horizon. As in S. Yang and D. Yang [42], we assume that the duration of each maintenance activity is a linear function of the running time of the machine and is denoted by $m_i = \alpha + \beta t_i$, where $\alpha > 0$ is the basic time of maintenance activity, $\beta > 0$ is the deteriorating maintenance factor, and t_i is the running time of the machine between the $(i-1)$th and ith maintenance activities of the machine. We further assume that a maintenance activity can be scheduled immediately after completing the processing of any job. After maintenance, the machine reverts to its initial condition and the deteriorating effect starts a new. The upper bound of the maintenance frequency on the machine due to the restriction of budget of maintenance is assumed to be known in advance. Denote by k and k_0 the maintenance frequency and the upper bound of the maintenance frequency on the machine, respectively, where $k \le k_0 \le n-1$.

There are three basic types of deteriorating effect investigated in our paper. The first type concerns the job-dependent deteriorating effect; that is, if job J_j is scheduled in the rth position, then its actual processing time p_{jr} is defined by

$$p_{jr} = p_j r^{a_j}, \quad j, r = 1, 2, \ldots, n, \tag{1}$$

where $a_j > 0$ is its deteriorating factor of job J_j.

The second type concerns the linear position-dependent deteriorating effect; that is, if job J_j is scheduled in the rth position, then its actual processing time p_{jr} is defined by

$$p_{jr} = p_j + b_j r, \quad j, r = 1, 2, \ldots, n, \tag{2}$$

where $b_j > 0$ is the deteriorating ratio of job J_j.

The third type concerns the linear time-dependent deteriorating effect model; that is, if job J_j is scheduled in the rth position, then its actual processing time p_{jr} is defined by

$$p_{jr} = p_j + cs_r, \quad j, r = 1, 2, \ldots, n, \tag{3}$$

where $c > 0$ is the common deteriorating factor and s_r is the starting time of a job processed in the position r in a sequence; that is, $s_1 = 0$.

The problem in this paper is to determine simultaneously the accepted job set A, the optimal maintenance frequencies, the optimal maintenance positions, and the optimal accepted job sequence S for minimizing the objective function

$$Z(S, A, Ma) = \sum_{j=1}^{n_a} C_{[j]} + \sum_{j \in \overline{A}} e_j \tag{4}$$

when the upper bound of the maintenance frequency on the machine is given in advance, where n_a is the number of accepted jobs and $C_{[j]}$ denotes the completion time of the jth job in the accepted job sequence. Using the standard three-field notation introduced by Graham et al. [48], our scheduling problem can be denoted as $1 \mid p_{jr}, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$, where p_{jr}, m_a, and k_0 in the second field represent the type of deteriorating effect, the maintenance activity, and the upper bound of the maintenance frequency on the machine, respectively.

3. Optimal Solution for the Case Where There Is No Maintenance

In this section we first study the problems under the case that there is no maintenance scheduled during the planning horizon (i.e., $k = 0$) and show that all the problems studied are polynomially solvable. We begin with providing a useful lemma which will be used later.

Lemma 1 (see [49]). *Consider two sequences of positive real numbers x_i and y_i $(i = 1, 2, \ldots, n)$. The value of $\sum_{i=1}^{n} x_i y_i$ is the least if the sequences are monotonic in the opposite sense.*

Since the objective function is regular, there is an optimal schedule for the problems under consideration in which all the accepted jobs are processed consecutively without idle time and the first job starts at time 0.

3.1. The Problem $1 \mid p_{jr}=p_j r^{a_j} \mid \sum_{j=1}^{n_a} C_{[j]}+\sum_{j\in\overline{A}} e_j$. This subsection addresses the problem of finding the accepted job set A and the optimal accepted job sequence to minimize the objective function given by (4) under model (1).

For a fixed job sequence $\pi = (S, \overline{A})$ with n_a accepted jobs in the order of S, where $S = (J_{[1]}, \ldots, J_{[n_a]})$, (4) can be reformulated as

$$Z\left(S, \overline{A}\right) = \sum_{r=1}^{n_a} \left(n_a - r + 1\right) p_{[r]} r^{a_{[r]}} + \sum_{j\in\overline{A}} e_j, \tag{5}$$

where $p_{[r]}$ and $a_{[r]}$ denote the normal processing time and aging factor of the job scheduled in the rth position in S, respectively.

Theorem 2. *Given the number of accepted jobs, the problem $1 \mid p_{jr} = p_j r^{a_j} \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$ can be formulated as an assignment problem.*

Proof. Let the number of accepted jobs be n_a $(1 \le n_a \le n)$. By (5), we can define

$$w_{jr}^{n_a} = \begin{cases} \left(n_a - r + 1\right) p_j r^{a_j} & r = 1, 2, \ldots, n_a, \\ e_j & r = n_a + 1, \ldots, n, \end{cases} \tag{6}$$

as the cost of assigning job $J_j (j = 1, 2, \ldots, n)$ to the rth $(r = 1, 2, \ldots, n)$ position in the schedule. Furthermore, let x_{jr} be the decision variable such that $x_{jr} = 1$ if job J_j is in the rth position, and $x_{jr} = 0$ otherwise, for $j, r = 1, 2, \ldots, n$. Then the scheduling problem can be formulated as the following assignment problem:

$$\begin{aligned} \text{Min} \quad & \sum_{j=1}^{n}\sum_{r=1}^{n} w_{jr}^{n_a} x_{jr}, \\ \text{s.t.} \quad & \sum_{r=1}^{n} x_{jr} = 1, \quad j = 1, 2, \ldots, n, \\ & \sum_{j=1}^{n} x_{jr} = 1, \quad r = 1, 2, \ldots, n, \\ & x_{jr} \in \{0, 1\}, \quad j, r = 1, 2, \ldots, n. \end{aligned} \tag{P1}$$

The result follows. □

Summing up the above analysis, the following solution algorithm can be presented for the problem $1 \mid p_{jr} = p_j r^{a_j} \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$.

Algorithm 3. Consider the following.

Step 1. For $n_a = 0$, calculate $F(0) = \sum_{j=1}^{n} e_j$.
Step 2. For $1 \le n_a \le n$, solve the above assignment problem (P1) and calculate the corresponding objective value $F(n_a)$.
Step 3. The optimal objective value is $\min\{F(n_a) \mid 0 \le n_a \le n\}$.

Theorem 4. *Algorithm 3 solves the problem $1 \mid p_{jr} = p_j r^{a_j} \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$ in $O(n^4)$ time.*

Proof. The correctness of Algorithm 3 follows directly from Theorem 2. Step 1 requires $O(n)$ time, while Step 2 needs to solve $n-1$ assignment problems where each one can be solved in $O(n^3)$ time. Therefore, the overall time complexity of the algorithm is $O(n^4)$. □

Now, let us turn our attention to the special case where $a_j = a$ for each $j = 1, 2, \ldots, n$, denoted as $1 \mid p_{jr} = p_j r^a \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$. In this case, (5) can be reformulated as follows.

$$Z\left(S, \overline{A}\right) = \sum_{r=1}^{n_a} \xi_r p_{[r]} + \sum_{j\in\overline{A}} e_j, \tag{7}$$

where $\xi_r = (n_a - r + 1) r^a$ for $r = 1, 2, \ldots, n$.

In what follows, we renumber the jobs in the nonincreasing order of their normal processing times (LPT order) and design a dynamic programming algorithm for the problem $1 \mid p_{jr} = p_j r^a \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$ as follows.

Let $(j, r)_{m_a}$ be a state, where m_a is the number of accepted jobs, j represents that the jobs $\{J_1, J_2, \ldots, J_j\}$ have been considered, and r, $0 \le r \le j$, represents how many of these jobs have been sequenced as accepted jobs, and let $f_{m_a}(j, r)$ be an optimal solution value for the state $(j, r)_{m_a}$, and let $S_{m_a}(j, r)$ be any schedule in state $(j, r)_{m_a}$ with solution value $f_{m_a}(j, r)$. By definition, set $f_{m_a}(j, r) = +\infty$ if no such schedule exists.

For given n_a, calculate each ξ_r values according to (7) for $r = 1, \ldots, n_a$. Sort ξ_r in nondecreasing order and reindex

them as $\xi_j^{n_a}$ for $j = 1, \ldots, n_a$ such that $\xi_1^{n_a} \le \xi_2^{n_a} \le \cdots \le \xi_{n_a}^{n_a}$. According to Lemma 1, the lth accepted job in $\{J_1, \ldots, J_j\}$ should be matched with $\xi_l^{n_a}$. Thus, the schedule $S_{m_a}(j, r)$ must have been constructed by taking one of the following two decisions in a previous state.

(1) J_j is scheduled as an accepted job: in this case $S_{m_a}(j, r)$ must have been obtained from schedule $S_{m_a}(j-1, r-1)$ and $f_{m_a}(j, r) = f_{m_a}(j-1, r-1) + p_j \xi_r^{n_a}$.

(2) J_j is scheduled as a rejected job: in this case $S_{m_a}(j, r)$ must have been obtained from schedule $S_{m_a}(j-1, r)$ and $f_{m_a}(j, r) = f_{m_a}(j-1, r) + e_j$.

Based on the above analysis, we can develop an algorithm as follows.

Algorithm 5. Consider the following.

Step 1. Renumber the jobs in the LPT order.
Step 2. For $0 \le n_a \le n$, we have the following.

Step 2.1. Calculate each ξ_r values according to (7) for $r = 1, \ldots, n_a$. Sort ξ_r in nondecreasing order and reindex them as $\xi_j^{n_a}$ for $j = 1, \ldots, n_a$ such that $\xi_1^{n_a} \le \xi_2^{n_a} \le \cdots \le \xi_{n_a}^{n_a}$. Initialize

$$f_{m_a}(j, r) = \begin{cases} 0 & \text{if } j = r = 0, \\ +\infty & \text{otherwise.} \end{cases} \tag{8}$$

Step 2.2. For $r = 0, 1, \ldots, \min\{m_a, j\}$, compute

$$f_{m_a}(j, r) = \min \begin{cases} f_{m_a}(j-1, r-1) + p_j \xi_r^{n_a} \\ f_{m_a}(j-1, r) + e_j. \end{cases} \tag{9}$$

Step 3. The optimal objective value is $F = \min\{f_{m_a}(n, m_a) \mid 0 \le m_a \le n\}$ and the resulting optimal accepted job sequence can be found by backtracking.

For each given n_a, the algorithm generates no more than n^2 states and computing each $f_{m_a}(j, r)$ requires constant time, so the time complexity is $O(n^3)$. This implies the following theorem.

Theorem 6. *Algorithm 5 solves the problem* $1 \mid p_{jr} = p_j r^a \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j \in \overline{A}} e_j$ *in* $O(n^3)$ *time.*

3.2. The Problem $1 \mid p_{jr} = p_j + b_j r \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j \in \overline{A}} e_j$. This subsection addresses the problem of finding the accepted job set A and the optimal accepted job sequence to minimize the objective function given by (4) under model (2).

For a fixed job sequence $\pi = (S, \overline{A})$ with n_a accepted jobs in the order of S, where $S = (J_{[1]}, \ldots, J_{[n_a]})$, (4) can be reformulated as

$$Z(S, \overline{A}) = \sum_{r=1}^{n_a} (n_a - r + 1)(p_{[r]} + r b_{[r]}) + \sum_{j \in \overline{A}} e_j. \tag{10}$$

Theorem 7. *Given the number of accepted jobs, the problem* $1 \mid p_{jr} = p_j + b_j r \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j \in \overline{A}} e_j$ *can be formulated as an assignment problem.*

Proof. Let the number of accepted jobs be n_a $(1 \le n_a \le n)$. By (5), we can define

$$\varepsilon_{jr}^{n_a} = \begin{cases} (n_a - r + 1)(p_j + r b_j) & r = 1, 2, \ldots, n_a, \\ e_j & r = n_a + 1, \ldots, n, \end{cases} \tag{11}$$

as the cost of assigning job J_j $(j = 1, 2, \ldots, n)$ to the rth $(r = 1, 2, \ldots, n)$ position in the schedule. Furthermore, let x_{jr} be the decision variable such that $x_{jr} = 1$ if job J_j is in the rth position, and $x_{jr} = 0$ otherwise, for $j, r = 1, 2, \ldots, n$. Then the scheduling problem can be formulated as the following assignment problem:

$$\begin{aligned} \text{Min} \quad & \sum_{j=1}^{n} \sum_{r=1}^{n} \varepsilon_{jr}^{n_a} x_{jr} \\ \text{s.t.} \quad & \sum_{r=1}^{n} x_{jr} = 1, \quad j = 1, 2, \ldots, n, \\ & \sum_{j=1}^{n} x_{jr} = 1, \quad r = 1, 2, \ldots, n, \\ & x_{jr} \in \{0, 1\}, \quad j, r = 1, 2, \ldots, n. \end{aligned} \tag{P2}$$

The result follows. □

Summing up the above analysis, the following solution algorithm can be presented for the problem $1 \mid p_{jr} = p_j + b_j r \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j \in \overline{A}} e_j$.

Algorithm 8. Consider the following.

Step 1. For $n_a = 0$, calculate $F(0) = \sum_{j=1}^{n} e_j$.
Step 2. For $1 \le n_a \le n$, solve the above assignment problem (P2) and calculate the corresponding objective value $F(n_a)$.
Step 3. The optimal objective value is $\min\{F(n_a) \mid 0 \le n_a \le n\}$.

Theorem 9. *Algorithm 8 solves the problem* $1 \mid p_{jr} = p_j + b_j r \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j \in \overline{A}} e_j$ *in* $O(n^4)$ *time.*

3.3. The Problem $1 \mid p_{jr} = p_j + c s_r \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j \in \overline{A}} e_j$. This subsection addresses the problem of finding the accepted job set A and the optimal accepted job sequence to minimize the objective function given by (4) under model (3).

For a fixed job sequence $\pi = (S, \overline{A})$ with n_a accepted jobs in the order of S, where $S = (J_{[1]}, \ldots, J_{[n_a]})$, (4) can be reformulated as

$$Z(S, \overline{A}) = \sum_{r=1}^{n_a} \eta_r p_{[r]} + \sum_{j \in \overline{A}} e_j, \tag{12}$$

where $\eta_r = \sum_{l=0}^{n_a - r} (1 + c)^l$. Since $c > 0$, we have that $\eta_1 > \eta_2 > \cdots > \eta_{n_a}$.

Define $\varphi_r = \sum_{l=0}^{r} (1 + c)^l$ for $r = 1, 2, \ldots, n$. Then $\varphi_1 < \varphi_2 < \cdots < \varphi_n$. Assume that the jobs have been reindexed in the LPT order. In what follows, we design a backward dynamic programming algorithm for the problem $1 \mid p_{jr} =$

$p_j + cs_r \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$. Let (j,r) be a state, in which j represents that the jobs $\{J_j, J_{j+1}, \ldots, J_n\}$ have been considered, and r, $0 \le r \le n-j+1$, represents how many of these jobs have been sequenced as accepted jobs, and let $f(j,r)$ be an optimal solution value for the state (j,r) and $S(j,r)$ any schedule in state (j,r) with solution value $f(j,r)$. By definition, set $f(j,r) = +\infty$ if no such schedule exists. Then, the schedule $S(j,r)$ must have been constructed by taking one of the following two decisions in a previous state.

(1) J_j is scheduled as an accepted job: in this case, by Lemma 1, the lth accepted job in $\{J_j, J_{j+1}, \ldots, J_n\}$ should be matched with φ_l; hence $S(j,r)$ must have been obtained from schedule $S(j+1, r-1)$ and $f(j,r) = f_{m_a}(j+1, r-1) + p_j\varphi_r$.

(2) J_j is scheduled as a rejected job: in this case $S(j,r)$ must have been obtained from schedule $S(j+1,r)$ and $f(j,r) = f(j+1,r) + e_j$.

The dynamic programming algorithm can be formally stated as follows.

Algorithm 10. Consider the following.

Step 1. Renumber the jobs in the LPT order.
Step 2. Initialize

$$f(j,r) = \begin{cases} 0 & \text{if } j = r = 0, \\ +\infty & \text{otherwise.} \end{cases} \tag{13}$$

Step 3. For $r = 0, 1, \ldots, j$, compute

$$f(j,r) = \min \begin{cases} f(j+1, r-1) + p_j\varphi_r \\ f(j+1, r) + e_j. \end{cases} \tag{14}$$

Step 4. The optimal objective value is $F = \min\{F(1,r) \mid 0 \le r \le n\}$ and the resulting optimal accepted job sequence can be found by backtracking.

Theorem 11. *Algorithm 10 solves the problem* $1 \mid p_{jr} = p_j + cs_r \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$ *in* $O(n^2)$ *time.*

4. Problems $1 \mid p_{jr}, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$

In this section we address the problem $1 \mid p_{jr}, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$ and show that all the problems consideration are polynomially solvable.

If the machine is subject to k times of maintenance, then there are $(k+1)$ groups of jobs in the accepted job sequence. Let $G_1, G_2, \ldots, G_{k+1}$ denote the groups of accepted jobs in the schedule. Then the group of accepted jobs and maintenance sequence can be denoted as $(G_1, M_1, G_2, M_2, \ldots, G_k, M_k, G_{k+1})$, where M_j represents the jth maintenance activity. Denote by $P(n, k+1) = (n_1, n_2, \ldots, n_{k+1})$ the allocation vector of the number of jobs in each group, where $n_j \ge 1$ is the number of jobs in group G_j and $\sum_{j=1}^{k+1} n_j = n$. Then group G_j can be denoted as $G_j = (J_{[j1]}, J_{[j2]}, \ldots, J_{[jn_j]})$.

Analogous to the analysis in Section 3, since the objective function is regular, there is an optimal schedule for the problems under consideration in which all the accepted jobs are processed consecutively without idle time and the first job starts at time 0.

4.1. The Problem $1 \mid p_{jr}=p_j r^{a_j}, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$. For a fixed job sequence and maintenance activities $\pi = (G_1, M_1, G_2, M_2, \ldots, G_k, M_k, G_{k+1}, \overline{A})$ with n_a accepted jobs, (4) can be reformulated as

$$Z\left(S, \overline{A}, MA\right) = \sum_{l=1}^{k}\left(n_a - \sum_{m=1}^{l} n_m\right)\alpha + \sum_{l=1}^{k+1}\sum_{r=1}^{n_l} w_r^{(l)} + \sum_{r=n_a+1}^{n} e_{[r]}, \tag{15}$$

where $w_r^{(l)} = [(n_l - r + 1 + n_a - \sum_{m=1}^{l} n_m) + (n_a - \sum_{m=1}^{l} n_m)\beta]\, p_{[lr]} r^{a_{[lr]}}$, $p_{[lr]}$ and $a_{[lr]}$ denote the normal processing time and deteriorating factor of the job scheduled in the rth position in l-group, respectively.

Theorem 12. *Given the number of accepted jobs, the problem* $1 \mid p_{jr} = p_j r^{a_j}, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$ *can be formulated as an assignment problem.*

Proof. Let the number of accepted jobs be n_a $(1 \le n_a \le n)$, and let x_{jr} be the decision variable such that $x_{jr} = 1$ if job J_j is in the rth position, and $x_{jr} = 0$ otherwise, for $j, r = 1, 2, \ldots, n$. Then the scheduling problem can be formulated as the following assignment problem:

$$\begin{aligned} \text{Min} \quad & \sum_{l=1}^{k}(n_a - y_l)\alpha \\ & + \sum_{j=1}^{n}\left(\sum_{l=1}^{k+1}\sum_{r=y_{l-1}}^{y_l}\left[(n_a - r + 1) + (n_a - y_l)\beta\right] \times p_j(r - y_{l-1})^{a_j} x_{jr} + \sum_{r=n_a+1}^{n} e_j x_{jr}\right), \\ \text{s.t.} \quad & \sum_{r=1}^{n} x_{jr} = 1, \qquad j = 1, 2, \ldots, n, \\ & \sum_{j=1}^{n} x_{jr} = 1, \qquad r = 1, 2, \ldots, n, \\ & x_{jr} \in \{0, 1\}, \qquad j, r = 1, 2, \ldots, n, \end{aligned} \tag{P3}$$

where $y_l = \sum_{m=1}^{l} n_m$, for $l = 1, \ldots, k+1$, and $y_0 = 0$. The result follows. □

Summing up the above analysis, the following solution algorithm can be presented for the problem $1 \mid p_{jr} = p_j r^{a_j}, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$.

Algorithm 13. Consider the following.

Step 1. For $n_a = 0$, calculate $f(0) = \sum_{j=1}^{n} e_j$.
Step 2. For given n_a, k, and vector $P(n_a, k+1) = (n_1, n_2, \ldots, n_{k+1})$, solve the above assignment problem (P3) and calculate the corresponding objective value $f(n_a, k, P(n_a, k+1))$.
Step 3. The optimal objective value is $\min\{f(n_a, k, P(n_a, k+1)) \mid 0 \le n_a \le n, 1 \le k \le n_a - 1, \sum_{j=1}^{k+1} n_j = n_a\}$.

Theorem 14. *Algorithm 13 solves the problem* $1 \mid p_{jr} = p_j r^{a_j}, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j \in \overline{A}} e_j$ *in* $O(n^{k_0+4})$ *time.*

Proof. The correction follows directly from Theorem 12. Step 1 requires $O(n)$ time. For given n_a, k, and $P(n_a, k+1)$, solving this assignment problem requires an effort of $O(n^3)$. Now, the question is how many $P(n_a, k+1)$ vectors exist. For given n_a and k, the number of jobs n_j in group G_j may be $1, \ldots, n_a - 1$, for $j = 1, \ldots, k+1$ and the number of jobs assigned to the last group (i.e., G_{k+1}) can be determined uniquely by the numbers of jobs on the first k groups. Thus we conclude that the upper bound of the number of $P(n_a, k+1)$ vectors is $(n_a - 1)^k \le n^k \le n^{k_0}$. Therefore, the overall time complexity of the algorithm is indeed $O(n^{k_0+4})$. □

Now, let us turn our attention to the special case where $a_j = a$ for each $j = 1, 2, \ldots, n$, denoted as $1 \mid p_{jr} = p_j r^a, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j \in \overline{A}} e_j$. In this case, (4) can be reformulated as follows:

$$Z\left(S, \overline{A}, MA\right) = \sum_{l=1}^{k} \left(n_a - \sum_{m=1}^{l} n_m \right) \alpha + \sum_{l=1}^{k+1} \sum_{r=1}^{n_l} \xi_r^{(l)} p_{[lr]} + \sum_{r=n_a+1}^{n} e_{[r]}, \tag{16}$$

where

$$\xi_r^{(l)} = \left[\left(n_l - r + 1 + n_a - \sum_{m=1}^{l} n_m \right) + \left(n_a - \sum_{m=1}^{l} n_m \right) \beta \right] r^a, \tag{17}$$

for $l = 1, \ldots, k+1$ and $r = 1, \ldots, n_l$.

Assume that the jobs have been reindexed in the LPT order. In what follows, we develop a forward dynamic programming algorithm to solve the problem $1 \mid p_{jr} = p_j r^a, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j \in \overline{A}} e_j$. Let $(j, r)_{(m_a, k, P(m_a,k))}$ be a state, where j represents that the jobs $\{J_1, J_2, \ldots, J_j\}$ have been considered, and r, $0 \le r \le j$, represents how many of these jobs have been sequenced as accepted jobs, while in the final optimal schedule for the whole job set, it contains exactly m_a accepted jobs, k maintenance activities, and n_i jobs in group G_i for $i = 1, \ldots, k+1$. Let $f_{(m_a,k,P(m_a,k))}(j, r)$ be an optimal solution value for the state $(j, r)_{(m_a,k,P(m_a,k))}$ and $S_{(m_a,k,P(m_a,k))}(j, r)$ any schedule in state $(j, r)_{(m_a,k,P(m_a,k))}$ with solution value $f_{(m_a,k,P(m_a,k))}(j, r)$. By definition, set $f_{(m_a,k,P(m_a,k))}(j, r) = +\infty$ if no such schedule exists.

For given m_a, k, and $P(m_a, k)$, calculate each $\xi_r^{(l)}$ values according to (17) for $l = 1, \ldots, k+1$ and $r = 1, \ldots, n_l$. Sort $\xi_r^{(l)}$ in nondecreasing order and reindex them as $\xi_j^{(m_a,k,P(m_a,k))}$ for $j = 1, \ldots, n_a$ such that $\xi_1^{(m_a,k,P(m_a,k))} \le \xi_2^{(m_a,k,P(m_a,k))} \le \cdots \le \xi_{n_a}^{(m_a,k,P(m_a,k))}$. According to Lemma 1, the lth accepted job in $\{J_1, \ldots, J_j\}$ should be matched with $\xi_l^{(m_a,k,P(m_a,k))}$. Thus, the schedule $S_{(m_a,k,P(m_a,k))}(j, r)$ must have been constructed by taking one of the following two decisions in a previous state.

(1) Job J_j is scheduled as an accepted job. In this case, $S_{(m_a,k,P(m_a,k))}(j, r)$ must have been obtained from schedule $S_{(m_a,k,P(m_a,k))}(j-1, r-1)$ and

$$f_{(m_a,k,P(m_a,k))}(j, r) = f_{(m_a,k,P(m_a,k))}(j-1, r-1) + p_j \xi_r^{(m_a,k,P(m_a,k))}. \tag{18}$$

(2) Job J_j is scheduled as a rejected job. In this case, $S_{(m_a,k,P(m_a,k))}(j, r)$ must have been obtained from schedule $S_{(m_a,k,P(m_a,k))}(j-1, r)$ and

$$f_{(m_a,k,P(m_a,k))}(j, r) = f_{(m_a,k,P(m_a,k))}(j-1, r-1) + e_j. \tag{19}$$

Based on the above analysis, we can develop an algorithm as follows.

Algorithm 15. Consider the following.

Step 1. Reindex the jobs in the in the LPT order.
Step 2. For $0 \le m_a \le n$, $0 \le k \le \min\{k_0, m_a - 1\}$, and $P(m_a, k) = (n_1, \ldots, n_{k+1})$, we have the following.
Step 2.1. Calculate $\xi^{(l)}(r)$ for $l = 1, \ldots, k+1$ and $r = 1, \ldots, n_l$ according to (17). Sort $\xi_r^{(l)}$ in nondecreasing order and reindex them as $\xi_j^{(m_a,k,P(m_a,k))}$ for $j = 1, \ldots, n_a$ such that $\xi_1^{(m_a,k,P(m_a,k))} \le \xi_2^{(m_a,k,P(m_a,k))} \le \cdots \le \xi_{n_a}^{(m_a,k,P(m_a,k))}$.
Step 2.2. For $0 \le j \le n$, $0 \le r \le j$, do as follows.
Initialize

$$f_{(m_a,k,P(m_a,k))}(j, r) = \begin{cases} 0 & \text{if } j = r = 0, \\ +\infty & \text{otherwise,} \end{cases} \tag{20}$$

and compute

$$f_{(m_a,k,P(m_a,k))}(j, r) = \min \begin{cases} f_{(m_a,k,P(m_a,k))}(j-1, r-1) + p_j \xi_r^{(m_a,k,P(m_a,k))}, \\ f_{(m_a,k,P(m_a,k))}(j-1, r-1) + e_j. \end{cases} \tag{21}$$

Step 3. The optimal solution value is $\min\{f_{(m_a,k,P(m_a,k))}(n, m_a) + \sum_{l=1}^{k}(n_a - \sum_{m=1}^{l} n_m)\alpha \mid 0 \le m_a \le n, 0 \le k \le \min\{k_0, m_a - 1\}, \sum_{j=1}^{k+1} n_j = n_a\}$ and the optimal schedule can be found by backtracking.

Theorem 16. *Algorithm 15 solves the problem* $1 \mid p_{jr} = p_j r^a, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$ *in* $O(n^{k_0+3})$ *time.*

Proof. Optimality is guaranteed by the principles underlying dynamic programming. We now consider the computational complexity of the dynamic programming solution algorithm. For each given m_a and k, the number of vectors $P(l, m)$ can be approximated as n^{k_0}, while for each given l, m, and $P(l, m)$, we have at most n^2 states and computing each $f_{(m_a,k,P(m_a,k))}(j, r)$ requires constant time. Therefore, the overall time complexity of the algorithm is $O(n^{k_0+3})$. □

4.2. The Problem $1 \mid p_{jr}=p_j+b_j r, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]}+\sum_{j\in\overline{A}} e_j$. For a fixed job sequence and maintenance activities $\pi = (G_1, M_1, G_2, M_2, \ldots, G_k, M_k, G_{k+1}, \overline{A})$ with n_a accepted jobs in the order of S, (4) can be reformulated as

$$Z\left(S, \overline{A}, MA\right) = \sum_{l=1}^{k}\left(n_a - \sum_{m=1}^{l} n_m\right)\alpha + \sum_{l=1}^{k+1}\sum_{r=1}^{n_l} \varepsilon_r^{(l)}\left(p_{[lr]} + b_{[lr]} r\right) + \sum_{r=n_a+1}^{n} e_{[r]}, \tag{22}$$

where

$$\varepsilon_r^{(l)} = \left[\left(n_l - r + 1 + n_a - \sum_{m=1}^{l} n_m\right) + \left(n_a - \sum_{m=1}^{l} n_m\right)\beta\right] \tag{23}$$

for $l = 1, \ldots, k+1$ and $r = 1, \ldots, n_l$.

Theorem 17. *Given the number of accepted jobs, the problem* $1 \mid p_{jr} = p_j + b_j r, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$ *can be formulated as an assignment problem.*

Proof. Let the number of accepted jobs be n_a $(1 \le n_a \le n)$, and let x_{jr} be the decision variable such that $x_{jr} = 1$ if job J_j is in the rth position, and $x_{jr} = 0$ otherwise, for $j, r = 1, 2, \ldots, n$. Then the scheduling problem can be formulated as the following assignment problem:

$$\begin{aligned}
\text{Min}\quad & \sum_{l=1}^{k}\left(n_a - y_l\right)\alpha \\
& + \sum_{j=1}^{n}\left(\sum_{l=1}^{k+1}\sum_{r=y_{l-1}}^{y_l}\left[\left(n_a - r + 1\right) + \left(n_a - y_l\right)\beta\right] \times \left(p_j + b_j\left(r - y_{l-1}\right)\right) x_{jr} + \sum_{r=n_a+1}^{n} e_j x_{jr}\right), \\
\text{s.t.}\quad & \sum_{r=1}^{n} x_{jr} = 1, \quad j = 1, 2, \ldots, n, \\
& \sum_{j=1}^{n} x_{jr} = 1, \quad r = 1, 2, \ldots, n, \\
& x_{jr} \in \{0, 1\}, \quad j, r = 1, 2, \ldots, n,
\end{aligned} \tag{P4}$$

where $y_l = \sum_{m=1}^{l} n_m$, for $l = 1, \ldots, k+1$, and $y_0 = 0$. The result follows. □

Summing up the above analysis, the following solution algorithm can be presented for the problem $1 \mid p_{jr} = p_j + b_j r, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$.

Algorithm 18. Consider the following.

Step 1. For $n_a = 0$, calculate $F(0) = \sum_{j=1}^{n} e_j$.

Step 2. For given n_a, k, and vector $P(n_a, k+1) = (n_1, n_2, \ldots, n_{k+1})$, solve the above assignment problem (P4) and calculate the corresponding objective value $F(n_a, k, P(n_a, k+1))$.

Step 3. The optimal objective value is $\min\{F(n_a, k, P(n_a, k+1)) \mid 0 \le n_a \le n, 1 \le k \le n_a - 1, \sum_{j=1}^{k+1} n_j = n_a\}$.

Theorem 19. *Algorithm 18 solves the problem* $1 \mid p_{jr} = p_j + b_j r, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$ *in* $O(n^{k_0+4})$ *time.*

4.3. The Problem $1 \mid p_{jr}=p_j+cs_r, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]}+\sum_{j\in\overline{A}} e_j$. For a fixed job sequence and maintenance activities $\pi = (G_1, M_1, G_2, M_2, \ldots, G_k, M_k, G_{k+1}, \overline{A})$ with n_a accepted jobs in the order of S, (4) can be reformulated as

$$Z\left(S, \overline{A}, MA\right) = \sum_{l=1}^{k}\left(n_a - \sum_{m=1}^{l} n_m\right)\alpha + \sum_{l=1}^{k+1}\sum_{r=1}^{n_l} \eta_r^{(l)} p_{[lr]} + \sum_{r=n_a+1}^{n} e_{[r]}, \tag{24}$$

where

$$\eta_r^{(l)} = (1+\beta)\left(n_a - \sum_{m=1}^{l} n_m\right)(1+c)^{n_l - r} + \sum_{i=1}^{n_l - r}(1+c)^i \tag{25}$$

for $l = 1, \ldots, k+1$ and $r = 1, \ldots, n_l$.

Analogous to the problem $1 \mid p_{jr} = p_j r^{a_j}, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$, Algorithm 13 also can be used for solving the problem $1 \mid p_{jr} = p_j + cs_r, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$ by replacing $\xi_r^{(l)}$ with $\eta_r^{(l)}$.

Theorem 20. *The problem* $1 \mid p_{jr} = p_j + cs_r, m_a \le k_0 \mid \sum_{j=1}^{n_a} C_{[j]} + \sum_{j\in\overline{A}} e_j$ *can be solved in* $O(n^{k_0+3})$ *time.*

Table 1: The summary of the obtained results.

Objective function	Complexity	Maintenance	Reference
$1\left\|p_{jr}=p_j r^{a_j}\right\|\sum_{j=1}^{n_a} C_{[j]}+\sum_{j\in\overline{A}} e_j$	$O\left(n^4\right)$	Not scheduled	Theorem 4
$1\left\|p_{jr}=p_j r^{a}\right\|\sum_{j=1}^{n_a} C_{[j]}+\sum_{j\in\overline{A}} e_j$	$O\left(n^3\right)$	Not scheduled	Theorem 6
$1\left\|p_{jr}=p_j+b_j r\right\|\sum_{j=1}^{n_a} C_{[j]}+\sum_{j\in\overline{A}} e_j$	$O\left(n^3\right)$	Not scheduled	Theorem 9
$1\left\|p_{jr}=p_j+cs_r\right\|\sum_{j=1}^{n_a} C_{[j]}+\sum_{j\in\overline{A}} e_j$	$O\left(n^2\right)$	Not scheduled	Theorem 11
$1\left\|p_{jr}=p_j r^{a_j}, m_a\le k_0\right\|\sum_{j=1}^{n_a} C_{[j]}+\sum_{j\in\overline{A}} e_j$	$O\left(n^{k_0+4}\right)$	Scheduled	Theorem 14
$1\left\|p_{jr}=p_j r^{a}, m_a\le k_0\right\|\sum_{j=1}^{n_a} C_{[j]}+\sum_{j\in\overline{A}} e_j$	$O\left(n^{k_0+3}\right)$	Scheduled	Theorem 16
$1\left\|p_{jr}=p_j+b_j r, m_a\le k_0\right\|\sum_{j=1}^{n_a} C_{[j]}+\sum_{j\in\overline{A}} e_j$	$O\left(n^{k_0+4}\right)$	Scheduled	Theorem 19
$1\left\|p_{jr}=p_j+cs_r, m_a\le k_0\right\|\sum_{j=1}^{n_a} C_{[j]}+\sum_{j\in\overline{A}} e_j$	$O\left(n^{k_0+3}\right)$	Scheduled	Theorem 20

5. Conclusions

This paper studies the single-machine scheduling problems with simultaneous considerations of job rejection, deterioration effects, and deteriorating multimaintenance activities with the objective of finding jointly the accepted job set, the optimal maintenance frequencies, the optimal maintenance positions, and the optimal job sequences such that the cost function based on the total completion time and rejection penalty is minimized, in which the upper bound of the total maintenance frequencies on the machine is assumed to be known in advance. The main results obtained are summarized in Table 1. Future research may focus on studying other models of maintenance duration, in multi-machine settings, and optimizing other performance measures.

References

[1] 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.

[2] T. C. E. Cheng, C. Wu, J. Chen, W. Wu, and S. Cheng, "Two-machine flowshop scheduling with a truncated learning function to minimize the makespan," *International Journal of Production Economics*, vol. 141, no. 1, pp. 79–86, 2013.

[3] T. C. E. Cheng, W. H. Kuo, and D. L. Yang, "Scheduling with a position-weighted learning effect based on sum-of-logarithm-processing-times and job position," *Information Sciences*, vol. 221, pp. 490–500, 2013.

[4] A. Janiak and R. Rudek, "A note on the learning effect in multi-agent optimization," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5974–5980, 2011.

[5] 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.

[6] A. Janiak and R. Rudek, "Experience-based approach to scheduling problems with the learning effect," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 39, no. 2, pp. 344–357, 2009.

[7] W. Kuo, C. Hsu, and D. Yang, "Worst-case and numerical analysis of heuristic algorithms for flowshop scheduling problems with a time-dependent learning effect," *Information Sciences*, vol. 184, no. 1, pp. 282–297, 2012.

[8] W. Lee and P. Lai, "Scheduling problems with general effects of deterioration and learning," *Information Sciences*, vol. 181, no. 6, pp. 1164–1170, 2011.

[9] R. Rudek, "The single processor total weighted completion time scheduling problem with the sum-of-processing-time based learning model," *Information Sciences*, vol. 199, pp. 216–229, 2012.

[10] 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.

[11] Y. Yin, D. Xu, and X. Huang, "Notes on 'some single-machine scheduling problems with general position-dependent and time-dependent learning effects'," *Information Sciences*, vol. 181, no. 11, pp. 2209–2217, 2011.

[12] A. Bachman, T. C. E. Cheng, A. Janiak, and C. T. Ng, "Scheduling start time dependent jobs to minimize the total weighted completion time," *Journal of the Operational Research Society*, vol. 53, no. 6, pp. 688–693, 2002.

[13] A. Bachman, T. C. E. Cheng, A. Janiak, and C. T. Ng, "Three scheduling problems with deteriorating jobs to minimize the total completion time," *Information Processing Letters*, vol. 81, no. 6, pp. 327–333, 2002.

[14] K. Inderfurth, A. Janiak, M. Y. Kovalyov, and F. Werner, "Batching work and rework processes with limited deterioration of reworkables," *Computers and Operations Research*, vol. 33, no. 6, pp. 1595–1605, 2006.

[15] A. Janiak and M. Y. Kovalyov, "Scheduling deteriorating jobs," in *Scheduling in Computer and Manufacturing Systems*, A. Janiak, Ed., pp. 12–25, WKL, Warszawa, Poland, 2006.

[16] M. M. Mazdeh, F. Zaerpour, A. Zareei, and A. Hajinezhad, "Parallel machines scheduling to minimize job tardiness and machine deteriorating cost with deteriorating jobs," *Applied Mathematical Modelling*, vol. 34, no. 6, pp. 1498–1510, 2010.

[17] Y. Yin, S. Cheng, and 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.

[18] Y. Yin, T. C. E. Cheng, J. Xu, S. R. Cheng, and C. C. Wu, "Single-machine scheduling with past-sequence-dependent delivery times and a linear deterioration," *Journal of Industrial and Management Optimization*, vol. 9, no. 2, pp. 323–339, 2013.

[19] Y. Yin, M. Liu, J. Hao, and M. Zhou, "Single-machine scheduling with job-position-dependent learning and time-dependent deterioration," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 42, no. 1, pp. 192–200, 2012.

[20] C. Zhao and H. Tang, "Rescheduling problems with deteriorating jobs under disruptions," *Applied Mathematical Modelling*, vol. 34, no. 1, pp. 238–243, 2010.

[21] T. C. E. Cheng, S. Yang, and D. Yang, "Common due-window assignment and scheduling of linear time-dependent deteriorating jobs and a deteriorating maintenance activity," *International Journal of Production Economics*, vol. 135, no. 1, pp. 154–161, 2012.

[22] J. Wang, "Single-machine scheduling problems with the effects of learning and deterioration," *Omega*, vol. 35, no. 4, pp. 397–402, 2007.

[23] J.-B. Wang, "A note on scheduling problems with learning effect and deteriorating jobs," *International Journal of Systems Science*, vol. 37, no. 12, pp. 827–832, 2006.

[24] J. Wang and T. C. E. Cheng, "Scheduling problems with the effects of deterioration and learning," *Asia-Pacific Journal of Operational Research*, vol. 24, no. 2, pp. 245–261, 2007.

[25] J. Wang, X. Huang, X. Wang, N. Yin, and L. Wang, "Learning effect and deteriorating jobs in the single machine scheduling problems," *Applied Mathematical Modelling*, vol. 33, no. 10, pp. 3848–3853, 2009.

[26] J. Wang, Y. Jiang, and G. Wang, "Single-machine scheduling with past-sequence-dependent setup times and effects of deterioration and learning," *International Journal of Advanced Manufacturing Technology*, vol. 41, no. 11-12, pp. 1221–1226, 2009.

[27] L. Wang, J. Wang, W. Gao, X. Huang, and E. Feng, "Two single-machine scheduling problems with the effects of deterioration and learning," *International Journal of Advanced Manufacturing Technology*, vol. 46, no. 5–8, pp. 715–720, 2010.

[28] D. Yang and W. Kuo, "Scheduling with deteriorating jobs and learning effects," *Applied Mathematics and Computation*, vol. 218, no. 5, pp. 2069–2073, 2011.

[29] Y. Yin and D. Xu, "Some single-machine scheduling problems with general effects of learning and deterioration," *Computers and Mathematics with Applications*, vol. 61, no. 1, pp. 100–108, 2011.

[30] B. Alidaee and N. K. Womer, "Scheduling with time dependent processing times: review and extensions," *Journal of the Operational Research Society*, vol. 50, no. 7, pp. 711–729, 1999.

[31] 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.

[32] T. C. E. Cheng, Q. Ding, and B. M. T. Lin, "A concise survey of scheduling with time-dependent processing times," *European Journal of Operational Research*, vol. 152, no. 1, pp. 1–13, 2004.

[33] G. Schmidt, "Scheduling with limited machine availability," *European Journal of Operational Research*, vol. 121, no. 1, pp. 1–15, 2000.

[34] Y. Ma, C. Chu, and C. Zuo, "A survey of scheduling with deterministic machine availability constraints," *Computers and Industrial Engineering*, vol. 58, no. 2, pp. 199–211, 2010.

[35] S. Gawiejnowicz, "Scheduling deteriorating jobs subject to job or machine availability constraints," *European Journal of Operational Research*, vol. 180, no. 1, pp. 472–478, 2007.

[36] M. Ji, Y. He, and T. C. E. Cheng, "Scheduling linear deteriorating jobs with an availability constraint on a single machine," *Theoretical Computer Science*, vol. 362, no. 1–3, pp. 115–126, 2006.

[37] W. Lee and C. Wu, "Multi-machine scheduling with deteriorating jobs and scheduled maintenance," *Applied Mathematical Modelling*, vol. 32, no. 3, pp. 362–373, 2008.

[38] C. Low, C. Hsu, and C. Su, "Minimizing the makespan with an availability constraint on a single machine under simple linear deterioration," *Computers and Mathematics with Applications*, vol. 56, no. 1, pp. 257–265, 2008.

[39] E. J. Lodree and C. D. Geiger, "A note on the optimal sequence position for a rate-modifying activity under simple linear deterioration," *European Journal of Operational Research*, vol. 201, no. 2, pp. 644–648, 2010.

[40] C. Wu and W. Lee, "Scheduling linear deteriorating jobs to minimize makespan with an availability constraint on a single machine," *Information Processing Letters*, vol. 87, no. 2, pp. 89–93, 2003.

[41] K. Rustogi and V. A. Strusevich, "Single machine scheduling with general positional deterioration and rate-modifying maintenance," *Omega*, vol. 40, no. 6, pp. 791–804, 2012.

[42] S. Yang and D. Yang, "Minimizing the total completion time in single-machine scheduling with aging/deteriorating effects and deteriorating maintenance activities," *Computers and Mathematics with Applications*, vol. 60, no. 7, pp. 2161–2169, 2010.

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

[44] S. Yang, D. Yang, and T. C. E. Cheng, "Single-machine due-window assignment and scheduling with job-dependent aging effects and deteriorating maintenance," *Computers and Operations Research*, vol. 37, no. 8, pp. 1510–1514, 2010.

[45] C. Zhao and H. Tang, "Single machine scheduling with general job-dependent aging effect and maintenance activities to minimize makespan," *Applied Mathematical Modelling*, vol. 34, no. 3, pp. 837–841, 2010.

[46] D. Shabtay, N. Gaspar, and M. Kaspi, "A survey on offline scheduling with rejection," *Journal of Scheduling*, vol. 16, pp. 3–28, 2013.

[47] Y. Bartal, S. Leonardi, A. Marchetti-Spaccamela, J. Sgall, and L. Stougie, "Multiprocessor scheduling with rejection," in *Proceedings of the 7th ACM-SIAM Symposium on Discrete Algorithms*, pp. 95–103, 2000.

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

[49] G. H. Hardy, J. E. Littlewood, and G. Polya, *Inequalities*, Cambridge University Press, London, UK, 1967.