Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 758919, 9 pages
http://dx.doi.org/10.1155/2015/758919

Research Article

Single-Machine Group Scheduling Problems with Variable Job Processing Times

Ping Ji[1] and Lin Li[2]

[1]*Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong*
[2]*College of Science, Shenyang Aerospace University, Shenyang, Liaoning, China*

Correspondence should be addressed to Ping Ji; p.ji@polyu.edu.hk

Received 2 September 2014; Accepted 16 November 2014

Academic Editor: Yunqiang Yin

This paper considers two resource constrained single-machine group scheduling problems. These problems involve variable job processing times (general position-dependent learning effects and deteriorating jobs); that is, the processing time of a job is defined by the function that involves its starting time and position in the group, and groups' setup time is a positive strictly decreasing continuous function of the amount of consumed resource. Polynomial time algorithms are proposed to optimally solve the makespan minimization problem under the constraint that the total resource consumption does not exceed a given limit and the total resource consumption minimization problem under the constraint that the makespan does not exceed a given limit, respectively.

1. Introduction

In classical scheduling theory, the jobs' processing time is always assumed to be fixed and constant value. Actually, we often encounter the case that job's processing time may be subject to change due to its position in the production sequence. Many authors have researched this aspect. Mosheiov [1] researched scheduling problems with a learning effect; that is, the production time of a given product is shorter if it is scheduled later rather than earlier in the sequence. Voutsinas and Pappis [2] discussed scheduling jobs with production times deteriorating exponentially over time in remanufacturing environments. The objective was to maximize total value of jobs. Cheng et al. [3] considered some scheduling problems with deteriorating jobs and learning effects. They derived polynomial time optimal solutions for the problems to minimize makespan and total completion time. Sun [4] considered a scheduling model in which deteriorating jobs and learning effect were both considered. The jobs' actual processing time depended on the processing time of the jobs already processed and their positions. He showed that the problems of minimizing the makespan, total completion time, and the sum of the quadratic job completion times remain polynomially solvable. Toksarı [5] presented a single-machine problem with the unequal release times under learning effect and deteriorating jobs. The objective was to minimize the makespan. He developed a branch-and-bound algorithm to solve it and proposed a heuristic algorithm to obtain a near optimal solution. Wang et al. [6] considered a single machine scheduling problem with a time-dependent learning effect and deteriorating jobs. The objective was to determine an optimal schedule so as to minimize the total completion time. They proposed three heuristic algorithms by using the V-shaped property. Wang et al. [7] studied a single-machine scheduling and due window assignment problem, which included deteriorating and learning effect. The objective was to minimize costs for earliness, tardiness, the window location, window size and makespan. They showed that the problem was polynomially solvable. Low and Lin [8] considered the model that the processing time of a job was determined by a function of its starting time and positional sequence in each machine.

They discussed different objectives, such as the makespan, total completion time, and weighted completion time. Single-machine scheduling problems could be solved polynomially. They also showed that flow shop scheduling problems were polynomially solvable under some certain conditions. Wang et al. [9] considered flowshop scheduling problems with a general exponential learning effect. The objective was to minimize the makespan, the total (weighted) completion time, the total weighted discounted completion time, and the sum of the quadratic job completion times, respectively. They proposed several simple heuristic algorithms to solve these problems and gave the tight worst-case bound of these heuristic algorithms.

In manufacturing industry, firms can increase production efficiency by using group technology (GT). GT is an approach in manufacturing that seeks to improve efficiency in high-volume production by exploiting the similarities of different products and activities in their production. Many researchers have paid great attention to this area and done a lot of work. Kuo and Yang [10] introduced a time-dependent learning effect into single-machine group scheduling problems. The time-dependent learning effect of a job is assumed to be a function of total processing times of jobs scheduled in front of it. They showed that the single-machine group scheduling problem with a time-dependent learning effect remains polynomially solvable for two objectives: minimizing the makespan and the total completion time. Wu et al. [11] considered two single-machine scheduling problems in the context of group technology where job processing times and setup times are simple linear functions of their starting times. The objectives were minimization of makespan and total completion time. Yan et al. [12] studied the single-machine scheduling problem with the effects of deterioration and learning, where the jobs were under group consumption and the setup time of each group was dependent on the resource it consumed. Zhang and Yan [13] introduced a deteriorated and learning effect into a single-machine problem where the learning effect not only depends on job position, but also depends on the group position; the deteriorated effect depends on its starting time of the job. They showed that the makespan, the total completion time, and maximum lateness problems remained polynomially optimally solvable under the proposed model. S. J. Yang and D. L. Yang [14] investigated single-machine group scheduling problems. The group setup time was assumed to follow a simple linear time-dependent deteriorating model. They examined two models of learning for the job processing time, provided polynomial time solutions for the makespan minimization problems, and showed that the total completion time minimization problems remained polynomially solvable under agreeable conditions. Zhu et al. [15] studied group scheduling problems with learning effects and resource allocation concurrently. They determined an optimal group sequence, internal job sequence, and the amount of resource allocated for two objectives, respectively. One was to minimize the weighted sum of makespan and total resource cost. The other was to minimize the weighted sum of total completion time and total resource cost. Lee and Lu [16] considered a single-machine scheduling problem in the context of group technology where the job processing times and the group setup times are simple linear functions of their starting times. The objective is to minimize the total weighted number of late jobs. They provided a branch-and-bound algorithm to solve the problem. Huang et al. [17] and Huang and Wang [18] considered two group scheduling problems with time and position-dependent processing times. Ji et al. [19] discussed group scheduling and job-dependent due window assignment problem and provided a polynomial algorithm to solve it. More recent papers have considered scheduling jobs with deteriorating jobs and/or learning effects including Wu et al. [11], Wu and Lee [20], Wang et al. [21], Wang [22], Yin et al. [23], Wang et al. [24], Lee and Wu [25], Li et al. [26], Zhu et al. [27], Huang et al. [28], Janiak and Rudek [29], Ji and Cheng [30], Yang [31], Yin and Xu [32], Bai et al. [33], Yin et al. [34], Yin et al. [35], J. B. Wang and J. J. Wang [36], Wu et al. [37], and Xu et al. [38].

Yan et al. [12] and Huang and Wang [18] considered the single-machine scheduling problems with the effects of learning and/or deterioration under the group technology consumption. They proved that the single-machine makespan minimization problem and total resource minimization problem under the group technology consumption remain polynomially solvable. This paper extends the results of Yan et al. [12] and Huang and Wang [18], by considering more general position-dependent learning effects and deteriorating jobs model that includes the one given in Yan et al. [12] and Huang and Wang [18] as a special case. The remaining part of this paper is organized as follows. In Section 2 we describe and formulate the problem. In Sections 3 and 4 we consider two resource-dependent scheduling problems and present polynomial algorithms to solve them. The last section includes conclusions.

2. Problem Formulation

There are n jobs to be processed on a single-machine, and these n jobs grouped into m groups. A setup time is required if the machine switches from one group to another and all setup times of groups for processing are at time 0. The machine can handle one job at a time and job preemption is not allowed. Let J_{ij} denote the jth job in group G_i, $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n_i$, where n_i denote the number of jobs belonging to group G_i (i.e., $n_1+n_2+\cdots+n_m = n$). Let s_i be the (actual) setup time of group G_i, and let $p_{ijr}(t)$ be the (actual) processing time of job J_{ij} if it is started at time t and scheduled in position r. In this paper, we consider the following model:

$$p_{ijr}(t) = p_{ij}(A \pm Bt) f_i(r), \tag{1}$$

where p_{ij} denotes the normal processing time for job J_{ij}, $A \geq 0, B \geq 0$. $f_i(r) \geq f_i(r+1)$ is the general case of positional learning for group G_i, specially $f_i(r) = r^{a_i}$ ($a_i \leq 0$ is the polynomial learning index for group G_i) and $f_i(r) = b_i^{r-1}$ ($0 < b_i \leq 1$ is the exponential learning index for group G_i). In addition, the setup time s_i is

$$s_i = g(u_i), \quad \underline{u} \leq u_i \leq \overline{u}, \ i = 1, 2, \ldots, m, \tag{2}$$

where u_i is the amount of a nonrenewable resource allocated to group G_i, with $0 \le u_i \le \overline{u}$, $\overline{u}$ is a technological constraint, and $g : R_+ \to R_+$ (R_+ is the set of nonnegative real numbers) is a strictly decreasing continuous function with $g(\overline{u}) \ge 0$. Let $u = (u_1, u_2, \ldots, u_n)$ be the resource allocation vector satisfying the resource constraint $\sum_{j=1}^{n} u_j \le U$, where U is the total amount of the resource available.

For a given schedule π, let $C_{ij} = C_{ij}(\pi)$ be the completion time of job J_{ij} under π, and let $C_{\max} = \max\{C_{ij} \mid i = 1, 2, \ldots, m; j = 1, 2, \ldots, n_i\}$ be the makespan (the maximal completion time) of a given schedule. We consider two problems: (1) minimizing the makespan with a resource consumption constraint and (2) the total resource consumption with a makespan constraint. Using the three-field notation (Graham [39]) for scheduling problems, these two problems can be denoted by $1 \mid s_i = g(u_i), p_{ijr}(t) = p_{ij}(A \pm Bt) f_i(r), GT, \sum_{j=1}^{n} u_j \le U \mid C_{\max}$ and $1 \mid s_i = g(u_i), p_{ijr}(t) = p_{ij}(A \pm Bt) f_i(r), GT, C_{\max} \le C \mid \sum u_j$.

3. The Problem $1 \mid s_i = g(u_i), p_{ijr}(t) = p_{ij}(A \pm Bt) f_i(r), GT, \sum_{j=1}^{n} u_j \le U \mid C_{\max}$

Lemma 1. *For a given schedule $\pi = [J_{[1]}, J_{[2]}, \ldots, J_{[n]}]$, if the first job starts at time $t_0 \ge 0$, then the completion time C_j is equal to*

$$C_{[j]} = \left(t_0 \pm \frac{A}{B}\right) \prod_{i=1}^{j} \left(1 \pm B p_{[i]} f(i)\right) \mp \frac{A}{B}, \tag{3}$$

where $[j] = i$ means job J_i is the jth one to be processed.

Proof (by mathematical induction method). For job $J_{[1]}$, we have

$$\begin{aligned} C_{[1]} &= t_0 + p_{[1]}\left(A \pm B t_0\right) f(1) \\ &= \left(t_0 \pm \frac{A}{B}\right)\left(1 \pm B p_1 f(1)\right) \mp \frac{A}{B}. \end{aligned} \tag{4}$$

Suppose Lemma 1 holds for job $J_{[j]}$; that is,

$$C_{[j]} = \left(t_0 \pm \frac{A}{B}\right) \prod_{i=1}^{j} \left(1 \pm B p_{[i]} f(i)\right) \mp \frac{A}{B}. \tag{5}$$

Then, for job $J_{[j+1]}$, we have

$$\begin{aligned} C_{[j+1]} &= C_{[j]} + p_{[j+1]}\left(A \pm B C_{[j]}\right) f(j+1) \\ &= \left(t_0 \pm \frac{A}{B}\right) \prod_{i=1}^{j+1} \left(1 \pm B p_{[i]} f(i)\right) \mp \frac{A}{B}. \end{aligned} \tag{6}$$

Hence, Lemma 1 holds. □

Theorem 2. *For the $1 \mid s_i = g(u_i), p_{ijr}(t) = p_{ij}(A \pm Bt) f_i(r), GT \mid C_{\max}$ problem, and a given schedule $\pi = [G_{[1]}, G_{[2]}, \ldots, G_{[m]}] = [J_{[1][1]}, J_{[1][2]}, \ldots, J_{[1][n_1]}, J_{[2][1]}, J_{[2][2]}, \ldots, J_{[2][n_2]}, \ldots, J_{[m][1]}, J_{[m][2]}, \ldots, J_{[m][n_m]}]$, the makespan is equal to*

$$\begin{aligned} C_{\max} &= C_{[m][n_m]} \\ &= \sum_{i=1}^{m} \left(s_{[i]} \prod_{l=i}^{m} \prod_{j=1}^{n_l} \left(1 \pm B p_{[l][j]} f_{[l]}(j)\right) \right) \\ &\quad \pm \frac{A}{B} \prod_{i=1}^{m} \prod_{j=1}^{n_i} \left(1 \pm B p_{[i][j]} f_{[i]}(j)\right) \mp \frac{A}{B}. \end{aligned} \tag{7}$$

Proof (by mathematical induction method). Stemming from Lemma 1, for group $G_{[1]}$, we have

$$C_{[1][n_1]} = \left(s_{[1]} \pm \frac{A}{B}\right) \prod_{j=1}^{n_1} \left(1 \pm B p_{[1][j]} f_{[1]}(j)\right) \mp \frac{A}{B}. \tag{8}$$

Suppose Theorem 2 holds for group $G_{[k]}$; that is,

$$\begin{aligned} C_{[k][n_k]} &= \sum_{i=1}^{k} \left(s_{[i]} \prod_{l=i}^{k} \prod_{j=1}^{n_l} \left(1 \pm B p_{[l][j]} f_{[l]}(j)\right) \right) \\ &\quad \pm \frac{A}{B} \prod_{i=1}^{k} \prod_{j=1}^{n_i} \left(1 \pm B p_{[i][j]} f_{[i]}(j)\right) \mp \frac{A}{B}. \end{aligned} \tag{9}$$

Then, for group $G_{[k+1]}$, we have

$$\begin{aligned} &C_{[k+1][n_{k+1}]} \\ &\quad = \left(C_{[k][n_k]} + s_{[k+1]} \pm \frac{A}{B}\right) \\ &\qquad \times \prod_{j=1}^{n_{k+1}} \left(1 \pm B p_{[k+1][j]} f_{[k+1]}(j)\right) \mp \frac{A}{B} \\ &\quad = \left(\sum_{i=1}^{k} \left(s_{[i]} \prod_{l=i}^{k} \prod_{j=1}^{n_l} \left(1 \pm B p_{[l][j]} f_{[l]}(j)\right) \right) \right. \\ &\qquad \left. \pm \frac{A}{B} \prod_{i=1}^{k} \prod_{j=1}^{n_i} \left(1 \pm B p_{[i][j]} f_{[i]}(j)\right) \mp \frac{A}{B} + s_{[k+1]} \pm \frac{A}{B} \right) \\ &\qquad \times \prod_{j=1}^{n_{k+1}} \left(1 \pm B p_{[k+1][j]} f_{[k+1]}(j)\right) \mp \frac{A}{B} \\ &\quad = \sum_{i=1}^{k+1} \left(s_{[i]} \prod_{l=i}^{k+1} \prod_{j=1}^{n_l} \left(1 \pm B p_{[l][j]} f_{[l]}(j)\right) \right) \\ &\qquad \pm \frac{A}{B} \prod_{i=1}^{k+1} \prod_{j=1}^{n_i} \left(1 \pm B p_{[i][j]} f_{[i]}(j)\right) \mp \frac{A}{B}. \end{aligned} \tag{10}$$

Hence, Theorem 2 holds. □

Theorem 3. *For the* $1 \mid s_i = g(u_i),\ p_{ijr}(t) = p_{ij}(A \pm Bt) f_i(r),\ GT \mid C_{\max}$ *problem, the optimal job sequence in each group can be obtained by the smallest processing time (SPT) first rule (i.e., arranging jobs in nondecreasing order of* p_{ij}*).*

Proof. It is similar to the proof of Bai et al. [33]. □

Theorem 4. *For the* $1 \mid s_i = g(u_i),\ p_{ijr}(t) = p_{ij}(A + Bt) f_i(r),\ GT \mid C_{\max}$ *problem, the groups scheduled earlier should be given the priority to resource allocation for any schedule of the groups.*

Proof. We assume $\pi = [G_{[1]}, G_{[2]}, \ldots, G_{[m]}]$ is any schedule of all the groups. Stemming from (7), first allocate the resource to the group, whose setup time $s_{[i]}$ is bigger than $\prod_{l=i}^{m}\prod_{j=1}^{n_l}(1+Bp_{[l][j]}f_{[l]}(j))$; the makespan can be minimized by the same amount of resource. Obviously

$$\begin{aligned}
&\prod_{l=1}^{m}\prod_{j=1}^{n_l}\left(1 + Bp_{[l][j]}f_{[l]}(j)\right)\\
&\quad\geq \prod_{l=2}^{m}\prod_{j=1}^{n_l}\left(1 + Bp_{[l][j]}f_{[l]}(j)\right)\\
&\quad\geq \cdots \geq \prod_{l=m}^{m}\prod_{j=1}^{n_l}\left(1 + Bp_{[l][j]}f_{[l]}(j)\right) \geq 1.
\end{aligned} \tag{11}$$

Hence, for the objective $C_{\max}$, the groups scheduled earlier should be given the priority to resource allocation. □

Theorem 5. *For the* $1 \mid s_i = g(u_i),\ p_{ijr}(t) = p_{ij}(A - Bt) f_i(r),\ GT \mid C_{\max}$ *problem, the groups scheduled later should be given the priority to resource allocation for any schedule of the groups.*

Proof. We assume $\pi = [G_{[1]}, G_{[2]}, \ldots, G_{[m]}]$ is any schedule of all the groups. Stemming from (7), first allocate the resource to the group, whose setup time $s_{[i]}$ is bigger than $\prod_{l=i}^{m}\prod_{j=1}^{n_l}(1-Bp_{[l][j]}f_{[l]}(j))$; the makespan can be minimized by the same amount of resource. Obviously

$$\begin{aligned}
&\prod_{l=1}^{m}\prod_{j=1}^{n_l}\left(1 - Bp_{[l][j]}f_{l}(j)\right)\\
&\quad\leq \prod_{l=2}^{m}\prod_{j=1}^{n_l}\left(1 - Bp_{[l][j]}f_{[l]}(j)\right)\\
&\quad\leq \cdots \leq \prod_{l=m}^{m}\prod_{j=1}^{n_l}\left(1 - Bp_{[l][j]}f_{[l]}(j)\right) \leq 1.
\end{aligned} \tag{12}$$

Hence, for the objective $C_{\max}$, the groups scheduled later should be given the priority to resource allocation. □

Theorem 6. *For the* $1 \mid s_i = g(u_i),\ p_{ijr}(t) = p_{ij}(A \pm Bt) f_i(r),\ GT \mid C_{\max}$ *problem, if resource amount of the groups in each position is fixed and in each group the jobs are scheduled by the SPT rule, then the optimal schedule can be obtained by scheduling the groups in nonincreasing order of* $\rho_{G_i} = \prod_{j=1}^{n_i}(1 \pm Bp_{ij}f_i(j))$.

Proof. Without loss of generality, we assume that the resource amount of the group scheduled at the ith position is denoted by $u_{[i]}$ $(i = 1, 2, \ldots, m)$. Let $\pi = [G_{[1]}, G_{[2]}, \ldots, G_{[v-1]}, G_v, G_{v+1}, G_{[v+2]}, \ldots, G_{[m]}]$ be a schedule and let $\pi' = [G_{[1]}, G_{[2]}, \ldots, G_{[v-1]}, G_{v+1}, G_v, G_{[v+2]}, \ldots, G_m]$ be a new schedule by changing the sequence of G_v and G_{v+1}. Obviously, for schedule π', $G_v(G_{v+1})$ is scheduled in the $(v+1)$th $(v$th$)$ position; then the resource allocation amount of $G_v(G_{v+1})$ is $u_{[v+1]}(u_{[v]})$. Then, under the schedules π and π', we have

$$\begin{aligned}
C_{\max}(\pi) = {}& f(u_{[1]})\prod_{l=1}^{m}\prod_{j=1}^{n_l}\left(1 \pm Bp_{[l][j]}f_{[l]}(j)\right)\\
&+ f(u_{[2]})\prod_{l=2}^{m}\prod_{j=1}^{n_l}\left(1 \pm Bp_{[l][j]}f_{[l]}(j)\right) + \cdots\\
&+ f(u_{[v-1]})\prod_{l=v-1}^{m}\prod_{j=1}^{n_l}\left(1 \pm Bp_{[l][j]}f_{[l]}(j)\right)\\
&+ f(u_{[v]})\prod_{l=v+2}^{m}\prod_{j=1}^{n_l}\left(1 \pm Bp_{[l][j]}f_{[l]}(j)\right)\\
&\times \prod_{j=1}^{n_v}\left(1 \pm Bp_{v[j]}f_{v}(j)\right)\\
&\times \prod_{j=1}^{n_{v+1}}\left(1 \pm Bp_{v+1[j]}f_{v+1}(j)\right)\\
&+ f(u_{[v+1]})\prod_{l=v+2}^{m}\prod_{j=1}^{n_l}\left(1 \pm Bp_{[l][j]}f_{[l]}(j)\right)\\
&\times \prod_{j=1}^{n_{v+1}}\left(1 \pm Bp_{v+1[j]}f_{v+1}(j)\right)\\
&+ f(u_{[v+2]})\prod_{l=v+2}^{m}\prod_{j=1}^{n_l}\left(1 \pm Bp_{[l][j]}f_{[l]}(j)\right) + \cdots\\
&+ f(u_{[m]})\prod_{l=m}^{m}\prod_{j=1}^{n_l}\left(1 \pm Bp_{[l][j]}f_{[l]}(j)\right)\\
&\pm \frac{A}{B}\prod_{i=1}^{m}\prod_{j=1}^{n_i}\left(1 \pm Bp_{[i][j]}f_{[i]}(j)\right) \mp \frac{A}{B},
\end{aligned}$$

$$\begin{aligned}
C_{\max}(\pi') = {}& f(u_{[1]})\prod_{l=1}^{m}\prod_{j=1}^{n_l}\left(1 \pm Bp_{[l][j]}f_{[l]}(j)\right)\\
&+ f(u_{[2]})\prod_{l=2}^{m}\prod_{j=1}^{n_l}\left(1 \pm Bp_{[l][j]}f_{[l]}(j)\right) + \cdots\\
&+ f(u_{[v-1]})\prod_{l=v-1}^{m}\prod_{j=1}^{n_l}\left(1 \pm Bp_{[l][j]}f_{[l]}(j)\right)
\end{aligned}$$

$$
\begin{aligned}
&+ f\left(u_{[v]}\right) \prod_{l=v+2}^{m} \prod_{j=1}^{n_l} \left(1 \pm Bp_{[l][j]} f_{[l]}(j)\right) \\
&\times \prod_{j=1}^{n_v} \left(1 \pm Bp_{v[j]} f_v(j)\right) \\
&\times \prod_{j=1}^{n_{v+1}} \left(1 \pm Bp_{v+1[j]} f_{v+1}(j)\right) \\
&+ f\left(u_{[v+1]}\right) \prod_{l=v+2}^{m} \prod_{j=1}^{n_l} \left(1 \pm Bp_{[l][j]} f_{[l]}(j)\right) \\
&\times \prod_{j=1}^{n_v} \left(1 \pm Bp_{v[j]} f_v(j)\right) \\
&\quad + f\left(u_{[v+2]}\right) \prod_{l=v+2}^{m} \prod_{j=1}^{n_l} \left(1 \pm Bp_{[l][j]} f_{[l]}(j)\right) + \cdots \\
&+ f\left(u_{[m]}\right) \prod_{l=m}^{m} \prod_{j=1}^{n_l} \left(1 \pm Bp_{[l][j]} f_{[l]}(j)\right) \\
&\pm \frac{A}{B} \prod_{i=1}^{m} \prod_{j=1}^{n_i} \left(1 \pm Bp_{[i][j]} f_{[i]}(j)\right) \mp \frac{A}{B}.
\end{aligned} \tag{13}
$$

Then

$$
\begin{aligned}
&C_{\max}(\pi) - C_{\max}\left(\pi'\right) \\
&= f\left(u_{[v+1]}\right) \prod_{l=v+2}^{m} \prod_{j=1}^{n_l} \left(1 \pm Bp_{[l][j]} f_{[l]}(j)\right) \\
&\quad \times \left(\prod_{j=1}^{n_{v+1}} \left(1 \pm Bp_{v+1[j]} f_{v+1}(j)\right) - \prod_{j=1}^{n_v} \left(1 \pm Bp_{v[j]} f_v(j)\right) \right) \\
&\le 0
\end{aligned} \tag{14}
$$

if and only if $\prod_{j=1}^{n_{v+1}}(1 \pm Bp_{v+1[j]} f_{v+1}(j)) \le \prod_{j=1}^{n_v}(1 \pm Bp_{v[j]} f_v(j))$. □

Based on Theorems 3, 4, and 6, an algorithm can be proposed to solve $1 \mid s_i = g(u_i), p_{ijr}(t) = p_{ij}(A + Bt) f_i(r), GT, \sum_{j=1}^{n} u_j \le U \mid C_{\max}$.

Algorithm 7. We have the following.

Step 1. Jobs in each group are scheduled by the SPT rule; that is,

$$p_{i[1]} \le p_{i[2]} \le \cdots \le p_{i[n_i]}, \quad i = 1, 2, \ldots, m. \tag{15}$$

Step 2. Groups are scheduled in nonincreasing order of $\rho_{G_{[i]}} = \prod_{j=1}^{n_i}(1 + Bp_{[i][j]} f_{[i]}(j))$; that is,

$$\rho_{G_{[1]}} \ge \rho_{G_{[2]}} \ge \cdots \ge \rho_{G_{[m]}}. \tag{16}$$

Table 1: The date of Example 9.

Group	G_1		G_2		G_3	
J_{ij}	J_{11}	J_{12}	J_{21}	J_{22}	J_{31}	J_{32}
p_{ij}	12	8	20	10	15	18
a_i	−0.2		−0.3		−0.1	

Step 3. For the obtained schedule $\pi = [G_{[1]}, G_{[2]}, \ldots, G_{[m]}]$, set $u^*_{[i]} = 0$ $(i = 1, 2, \ldots, m)$ and $l = 1$.

Step 4. Set $u^*_{[l]} = \min\{\overline{u}, U\}$, $U = U - u^*_{[l]}$, and $l = l + 1$.

Step 5. If $U = 0$ or $l = m + 1$, stop; else go to Step 4.

Based on Theorems 3, 5, and 6, an algorithm can be proposed to solve $1 \mid s_i = g(u_i), p_{ijr}(t) = p_{ij}(A - Bt) f_i(r), GT, \sum_{j=1}^{n} u_j \le U \mid C_{\max}$.

Algorithm 8. We have the following.

Step 1. Jobs in each group are scheduled by the SPT rule; that is,

$$p_{i[1]} \le p_{i[2]} \le \cdots \le p_{i[n_i]}, \quad i = 1, 2, \ldots, m. \tag{17}$$

Step 2. Groups are scheduled in nonincreasing order of $\rho_{G_{[i]}} = \prod_{j=1}^{n_i}(1 + Bp_{[i][j]} f_{[i]}(j))$; that is,

$$\rho_{G_{[1]}} \ge \rho_{G_{[2]}} \ge \cdots \ge \rho_{G_{[m]}}. \tag{18}$$

Step 3. For the obtained schedule $\pi = [G_{[1]}, G_{[2]}, \ldots, G_{[m]}]$, set $u^*_{[i]} = 0$ $(i = 1, 2, \ldots, m)$ and $l = m$.

Step 4. Set $u^*_{[l]} = \min\{\overline{u}, U\}$, $U = U - u^*_l$, and $l = l - 1$.

Step 5. If $U = 0$ or $l = 0$, stop; else go to Step 4.

Obviously the total time for Algorithm 7 (Algorithm 8) is $O(n \log n)$.

The following example is used to illustrate Algorithm 7.

Example 9. Consider $n = 6$, $m = 3$, $f_i(r) = r^{a_i}$, and $g(u_i) = 6 - u_i$. The normal processing times p_{ij}, a_i for each group are given in Table 1, $A = 1$, $B = 0.1$, $\overline{u} = 5$, and $U = 10$.

Solution. According to Algorithm 7, we solve Example 9 as follows.

Step 1. In group G_1, the optimal job sequence is $J_{12} \to J_{11}$. In group G_2, the optimal job sequence is $J_{22} \to J_{21}$. In group G_3, the optimal job sequence is $J_{31} \to J_{32}$.

Step 2. Consider $\rho(G_1) = 3.6804 < \rho(G_2) = 5.2490 < \rho(G_3) = 6.6987$.

Step 3. The obtained schedule is $\pi = [G_3, G_2, G_1]$.

Steps 4 and 5. The optimal resources $u^*_3 = \min\{\overline{u}, U\} = 5$, $u^*_2 = 5$, $u^*_1 = 0$.

Therefore, the optimal schedule is $[J_{31} \rightarrow J_{32}] \rightarrow [J_{22} \rightarrow J_{21}] \rightarrow [J_{12} \rightarrow J_{11}]$, and the optimal completion times are calculated as follows:

$$\begin{aligned}
s_3 &= 6 - 5 = 1,\\
C_{31} &= 1 + 15 * (1 + 0.1 * 1) * 1^{-0.1} = 17.5,\\
C_{32} &= 17.5 + 18 * (1 + 0.1 * 17.5) * 2^{-0.1} = 63.6851,\\
s_2 &= 6 - 5 = 1,\\
C_{22} &= 63.6851 + 1 + 10 * (1 + 0.1 * (63.6851 + 1)) * 1^{-0.3}\\
&= 139.3702,\\
C_{21} &= 139.3702 + 20 * (1 + 0.1 * 139.3702) * 2^{-0.3}\\
&= 382.0228,\\
s_1 &= 6 - 0 = 6,\\
C_{12} &= 382.0228 + 6 + 8 * (1 + 0.1 * (6 + 382.0228)) * 1^{-0.2}\\
&= 706.4410,\\
C_{11} &= 706.4410 + 12 * (1 + 0.1 * 706.4410) * 2^{-0.2}\\
&= 1454.879.
\end{aligned} \tag{19}$$

Hence $C_{\max} = C_{12} = 1454.879$.

4. The Problem $1 \mid s_i = g(u_i),\ p_{ijr}(t) = p_{ij}(A \pm Bt) f_i(r), GT, C_{\max} \le C \mid \sum u_j$

Similar to Section 3, we only need to consider the schedule in which the jobs in each group are scheduled by the SPT rule, and the groups scheduled in nonincreasing order of $\rho_{G_{[i]}} = \prod_{j=1}^{n_i}(1 \pm Bp_{[i][j]} f_{[i]}(j))$ and the groups scheduled earlier (later) should be given the priority to resource allocation for the problem $1 \mid s_i = g(u_i),\ p_{ijr}(t) = p_{ij}(A + Bt) f_i(r), GT, C_{\max} \le C \mid \sum u_j (1 \mid s_i = g(u_i), p_{ijr}(t) = p_{ij}(A - Bt) f_i(r), GT, C_{\max} \le C \mid \sum u_j)$.

For a given schedule $\pi = [G_{[1]}, G_{[2]}, \ldots, G_{[m]}] = [J_{[1][1]}, J_{[1][2]}, \ldots, J_{[1][n_1]}, J_{[2][1]}, J_{[2][2]}, \ldots, J_{[2][n_2]}, \ldots, J_{[m][1]}, J_{[m][2]}, \ldots, J_{[m][n_m]}]$, stemming from (7), the minimum makespan of this schedule is

$$\begin{aligned}
C_{\max} = &\sum_{i=1}^{m}\left(f(\overline{u}) \prod_{l=i}^{m}\prod_{j=1}^{n_l}\left(1 \pm Bp_{[l][j]} f_{[l]}(j)\right)\right)\\
&\pm \frac{A}{B}\prod_{i=1}^{m}\prod_{j=1}^{n_i}\left(1 \pm Bp_{[i][j]} f_{[i]}(j)\right) \mp \frac{A}{B}.
\end{aligned} \tag{20}$$

So, the schedule π is feasible only if

$$\begin{aligned}
C_{\max} = &\sum_{i=1}^{m}\left(f(\overline{u}) \prod_{l=i}^{m}\prod_{j=1}^{n_l}\left(1 \pm Bp_{[l][j]} f_{[l]}(j)\right)\right)\\
&\pm \frac{A}{B}\prod_{i=1}^{m}\prod_{j=1}^{n_i}\left(1 \pm Bp_{[i][j]} f_{[i]}(j)\right) \mp \frac{A}{B} \le C.
\end{aligned} \tag{21}$$

Since the problem is to minimize the total resource consumption with a makespan constraint, then resource consumption of groups should be given as few as possible and the completion time of the last group is C; that is, $C_{\max} = C$.

Without loss of generality, let $\pi = [G_{[1]}, G_{[2]}, \ldots, G_{[m]}] = [J_{[1][1]}, J_{[1][2]}, \ldots, J_{[1][n_1]}, J_{[2][1]}, J_{[2][2]}, \ldots, J_{[2][n_2]}, \ldots, J_{[m][1]}, J_{[m][2]}, \ldots, J_{[m][n_m]}]$ be the schedule in which the jobs in each group are scheduled by the SPT rule and the groups scheduled in nonincreasing order of $\rho_{G_{[i]}} = \prod_{j=1}^{n_i}(1 \pm Bp_{[l][j]} f_{[l]}(j))$.

For the problem $1 \mid s_i = g(u_i),\ p_{ijr}(t) = p_{ij}(A + Bt) f_i(r), GT, C_{\max} \le C \mid \sum u_j$, if we only allocate the resource to the setup time of group $G_{[1]}$ and for the groups scheduled after it the resource allocation amount is 0, then according to (7), we have

$$\begin{aligned}
C = &s_{[1]}\prod_{l=1}^{m}\prod_{j=1}^{n_l}\left(1 + Bp_{[l][j]} f_{[l]}(j)\right)\\
&+ s\sum_{i=2}^{m}\left(\prod_{l=i}^{m}\prod_{j=1}^{n_l}\left(1 + Bp_{[l][j]} f_{[l]}(j)\right)\right)\\
&+ \frac{A}{B}\prod_{i=1}^{m}\prod_{j=1}^{n_i}\left(1 + Bp_{[i][j]} f_{[i]}(j)\right) - \frac{A}{B}\\
= &s_{[1]}\prod_{l=1}^{m}\prod_{j=1}^{n_l}\left(1 + Bp_{[l][j]} f_{[l]}(j)\right)\\
&+ s\sum_{i=2}^{m}\left(\prod_{l=i}^{m}\prod_{j=1}^{n_l}\left(1 + Bp_{[l][j]} f_{[l]}(j)\right)\right) + K,
\end{aligned} \tag{22}$$

where $s = g(0)$ and $K = (A/B)\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1 + Bp_{[i][j]} f_{[i]}(j)) - A/B$. Hence

$$s_{[1]} = \frac{C - K - s\sum_{i=2}^{m}\left(\prod_{l=i}^{m}\prod_{j=1}^{n_l}\left(1 + Bp_{[l][j]} f_{[l]}(j)\right)\right)}{\prod_{l=1}^{m}\prod_{j=1}^{n_l}\left(1 + Bp_{[l][j]} f_{[l]}(j)\right)}. \tag{23}$$

If $s_{[1]} \ge g(0)$, then the constraint $C_{\max} \le C$ can be satisfied even if the resource allocation amount $u_{[1]}$ of $G_{[1]}$ is 0. And we do not have to allocate the resource to the groups; stop.

If $g(\overline{u}) \le s_{[1]} < g(0)$, then the optimal resource allocation is $u^*_{[1]} = g^{-1}(s_{[1]})$, $u^*_{[j]} = 0$, $j = 2, \ldots, n$.

If $s_{[1]} < g(\overline{u})$, then the optimal resource allocation of group $G_{[1]}$ should be $\overline{u}$; that is, $u^*_{[1]} = \overline{u}$. And the optimal resource consumption $u^*_{[j]}$, $j = 2, \ldots, n$, can be calculated by the same method.

Based on the above analysis, we propose an algorithm to solve the problem $1 \mid s_i = g(u_i), p_{ijr}(t) = p_{ij}(A + Bt)f_i(r), GT, C_{\max} \le C \mid \sum u_j$.

Algorithm 10. We have the following.

Step 1. Jobs in each group are scheduled in nondecreasing order of p_{ij}; that is,

$$p_{i[1]} \le p_{i[2]} \le \cdots \le p_{i[n_i]}, \quad i = 1, 2, \ldots, m. \tag{24}$$

Step 2. Groups are scheduled in nonincreasing order of $\rho_{G_{[i]}} = \prod_{j=1}^{n_i}(1 + Bp_{[i][j]}f_{[i]}(j))$; that is,

$$\rho_{G_{[1]}} \ge \rho_{G_{[2]}} \ge \cdots \ge \rho_{G_{[m]}}. \tag{25}$$

Step 3. For the obtained schedule $\pi = [G_{[1]}, G_{[2]}, \ldots, G_{[m]}]$, set $u^*_{[i]} = 0$ $(i = 1, 2, \ldots, m)$, $U = 0$, $s = g(0)$, $s' = g(\overline{u})$, $l = 1$, and $C' = C - K$.

Step 4. Calculate

$$s_{[l]} = \frac{C' - s\sum_{i=l+1}^{m}\left(\prod_{k=i}^{m}\prod_{j=1}^{n_k}\left(1 + Bp_{[k][j]}f_{[k]}(j)\right)\right)}{\prod_{i=l}^{m}\prod_{j=1}^{n_i}\left(1 + Bp_{[i][j]}f_{[i]}(j)\right)}. \tag{26}$$

If $l = m + 1$, stop; else go to Step 5.

Step 5. If $s_{[l]} \ge g(0)$, stop.

If $g(\overline{u}) \le s_{[l]} < g(0)$, set $u^*_l = g^{-1}(s_{[l]})$ and $U = U + u^*_l$ and stop; the optimal resource allocation is $u^* = [u^*_1, u^*_2, \ldots, u^*_m]$.

If $s_{[l]} < g(\overline{u})$, go to Step 6.

Step 6. Set $u^*_{[l]} = \overline{u}$, $U = U + u^*_{[l]}$, $C' = C' - s'\sum_{i=l}^{m}\prod_{j=1}^{n_i}(1 + Bp_{[i][j]}f_{[i]}(j))$, and $l = l + 1$. If $l > m$, then stop; there is no feasible solution; else go to Step 4.

For the problem $1 \mid s_i = g(u_i), p_{ijr}(t) = p_{ij}(A - Bt)f_i(r), GT, C_{\max} \le C \mid \sum u_j$, if we only allocate the resource to the setup time of group G_m and for the groups scheduled before it the resource allocation amount is 0, then according to (7), we have

$$\begin{aligned} C &= s\sum_{i=1}^{m-1}\left(\prod_{l=i}^{m}\prod_{j=1}^{n_l}\left(1 - Bp_{[l][j]}f_{[l]}(j)\right)\right) \\ &\quad + s_{[m]}\left(\prod_{j=1}^{n_m}\left(1 - Bp_{[m][j]}f_{[m]}(j)\right)\right) \\ &\quad - \frac{A}{B}\prod_{i=1}^{m}\prod_{j=1}^{n_i}\left(1 - Bp_{[i][j]}f_{[i]}(j)\right) + \frac{A}{B} \\ &= s\sum_{i=1}^{m-1}\left(\prod_{l=i}^{m}\prod_{j=1}^{n_l}\left(1 - Bp_{[l][j]}f_{[l]}(j)\right)\right) \\ &\quad + s_{[m]}\left(\prod_{j=1}^{n_m}\left(1 - Bp_{[m][j]}f_{[m]}(j)\right)\right) + K, \end{aligned} \tag{27}$$

where $s = g(0)$ and $K = A/B - (A/B)\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1 - Bp_{[i][j]}f_{[i]}(j))$. Hence

$$s_{[m]} = \frac{C - K - s\sum_{i=1}^{m-1}\left(\prod_{l=i}^{m}\prod_{j=1}^{n_l}\left(1 - Bp_{[l][j]}f_{[l]}(j)\right)\right)}{\prod_{j=1}^{n_m}\left(1 - Bp_{[m][j]}f_{[m]}(j)\right)}. \tag{28}$$

If $s_{[m]} \ge g(0)$, then the constraint $C_{\max} \le C$ can be satisfied even if the resource allocation amount $u_{[m]}$ of $G_{[m]}$ is 0. And we do not have to allocate the resource to the groups; stop.

If $g(\overline{u}) \le s_{[m]} < g(0)$, then the optimal resource allocation is $u^*_{[m]} = g^{-1}(s_{[m]})$, $u^*_{[j]} = 0$, $j = 1, 2, \ldots, m - 1$.

If $s_{[m]} < g(\overline{u})$, then the optimal resource allocation of group $G_{[m]}$ should be $\overline{u}$; that is, $u^*_{[m]} = \overline{u}$. And the optimal resource consumption $u^*_{[j]}$, $j = 1, 2, \ldots, m - 1$, can be calculated by the same method.

Based on the above analysis, we propose an algorithm to solve the problem $1 \mid s_i = g(u_i), p_{ijr}(t) = p_{ij}(A - Bt)f_i(r), GT, C_{\max} \le C \mid \sum u_j$.

Algorithm 11. We have the following.

Step 1. Jobs in each group are scheduled in nondecreasing order of p_{ij}; that is,

$$p_{i[1]} \le p_{i[2]} \le \cdots \le p_{i[n_i]}, \quad i = 1, 2, \ldots, m. \tag{29}$$

Step 2. Groups are scheduled in nonincreasing order of $\rho_{G_{[i]}} = \prod_{j=1}^{n_i}(1 - Bp_{[i][j]}f_{[i]}(j))$; that is,

$$\rho_{G_{[1]}} \ge \rho_{G_{[2]}} \ge \cdots \ge \rho_{G_{[m]}}. \tag{30}$$

Step 3. For the obtained schedule $\pi = [G_{[1]}, G_{[2]}, \ldots, G_{[m]}]$, set $u^*_{[i]} = 0$ $(i = 1, 2, \ldots, m)$, $U = 0$, $s = g(0)$, $s' = g(\overline{u})$, $l = m$, and $C' = C - K$.

Step 4. Calculate

$$s_{[l]} = \frac{C' - s\sum_{i=1}^{l-1}\left(\prod_{l=i}^{m}\prod_{j=1}^{n_l}\left(1 - Bp_{[l][j]}f_{[l]}(j)\right)\right)}{\prod_{i=l}^{m}\prod_{j=1}^{n_i}\left(1 - Bp_{[i][j]}f_{[i]}(j)\right)}. \tag{31}$$

If $l = 0$, stop; else go to Step 5.

Step 5. If $s_{[l]} \ge g(0)$, stop.

If $g(\overline{u}) \le s_{[l]} < g(0)$, set $u^*_{[l]} = g^{-1}(s_{[l]})$ and $U = U + u^*_{[l]}$ and stop; the optimal resource allocation is $u^* = [u^*_{[1]}, u^*_{[2]}, \ldots, u^*_{[m]}]$.

If $s_{[l]} < g(\overline{u})$, go to Step 6.

Step 6. Set $u^*_{[l]} = \overline{u}$, $U = U + u^*_{[l]}$, $C' = C' - s'\sum_{i=l}^{m}\prod_{j=1}^{n_i}(1 - Bp_{[i][j]}f_{[i]}(j))$, and $l = l - 1$. If $l < 1$, then stop; there is no feasible solution; else go to Step 4.

Obviously, the time for Steps 1 and 2 is $O(n \log n)$. If we are able to calculate all the values of g and g^{-1}

in Algorithm 10 in $O(h(n))$ time, then the total time for Algorithm 10 (Algorithm 11) is $O(\max\{nh(n), n\log n\})$ time.

The following example is used to illustrate Algorithm 10.

Example 12. Consider $n = 6$, $m = 3$, $f_i(r) = r^{a_i}$, and $g(u_i) = 26 - 2u_i$. The normal processing times p_{ij} and a_i for each group are the same as Table 1, $A = 1$, $B = 0.1$, $\overline{u} = 10$, and $C = 3000$.

Solution. According to Algorithm 10, we solve Example 12 as follows.

Steps 1, 2, and 3. Similar to Example 9, the obtained schedule is $[J_{31} \to J_{32}] \to [J_{22} \to J_{21}] \to [J_{12} \to J_{11}]$. Set $u_i^* = 0$ $(i = 1, 2, \ldots, m)$, $U = 0$, $s = g(0) = 26$, $s' = g(\overline{u}) = 6$, $l = 1$, $K = 1284.072$, and $C' = C - K = 3000 - 1284.072 = 1715.928$.

Step 4. Calculate

$$s_{[1]} = s_3 = \frac{C' - s\sum_{i=l+1}^{m}\left(\prod_{k=i}^{m}\prod_{j=1}^{n_k}\left(1 + Bp_{[k][j]}f_{[k]}(j)\right)\right)}{\prod_{i=l}^{m}\prod_{j=1}^{n_i}\left(1 + Bp_{[i][j]}f_{[i]}(j)\right)} = 8.6391. \tag{32}$$

Steps 5 and 6. Consider $6 < 8.6391 < 26$, $u_3^* = g^{-1}(8.6391) = 8.6805$, $U = U + u_3^* = 8.6805$, $C' = C' - s'\sum_{i=l}^{m}\prod_{j=1}^{n_i}(1 + Bp_{[i][j]}f_{[i]}(j)) = 1577.935$, and $l = 2$.

Similarly $s_{[2]} = s_2 = 76.72711 > 26$; stop. Hence, $u_3^* = 8.6805$, $u_2^* = u_1^* = 0$, and $U = 10$.

Therefore, the optimal schedule is $[J_{31} \to J_{32}] \to [J_{22} \to J_{21}] \to [J_{12} \to J_{11}]$, and the optimal completion times are calculated as follows:

$$\begin{aligned}
s_3 &= 8.6391,\\
C_{31} &= 8.6391 + 15 * (1 + 0.1 * 8.6391) * 1^{-0.1}\\
&= 36.5978,\\
C_{32} &= 36.5978 + 18 * (1 + 0.1 * 36.5978) * 2^{-0.1}\\
&= 114.8569,\\
s_2 &= 26,\\
C_{22} &= 114.8569 + 26\\
&\quad + 10 * (1 + 0.1 * (114.8569 + 26)) * 1^{-0.3}\\
&= 291.7138,\\
C_{21} &= 291.7138 + 20 * (1 + 0.1 * 291.7138) * 2^{-0.3}\\
&= 781.8493,\\
s_1 &= 26,\\
C_{12} &= 781.8493 + 26\\
&\quad + 8 * (1 + 0.1 * (781.8493 + 26)) * 1^{-0.2}\\
&= 1462.129,\\
C_{11} &= 1462.129 + 12 * (1 + 0.1 * 1462.129) * 2^{-0.2}\\
&\approx 3000.
\end{aligned} \tag{33}$$

5. Conclusions

We considered some single-machine group scheduling problems with general position-dependent learning effects and deteriorating jobs simultaneously. It is assumed that group setup time of a group is a positive strictly decreasing continuous function of the amount of consumed resource. For the resource constrained problems $1 \mid s_i = g(u_i), p_{ijr}(t) = p_{ij}(A \pm Bt)f_i(r), GT, \sum_{j=1}^{n} u_j \le U \mid C_{\max}$ and $1 \mid s_i = g(u_i), p_{ijr}(t) = p_{ij}(A \pm Bt)f_i(r), GT, C_{\max} \le C \mid \sum u_j$, we showed that they can be solved in polynomial time under the proposed model.

Conflict of Interests

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

Acknowledgments

This research was supported by a grant from the Research Grants Council of Hong Kong Special Administrative Region, China (PolyU 517011), and the National Natural Science Foundation of China (no. 61403260, no. 71471120).

References

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

[2] T. G. Voutsinas and C. P. Pappis, "Scheduling jobs with values exponentially deteriorating over time," *International Journal of Production Economics*, vol. 79, no. 3, pp. 163–169, 2002.

[3] T. C. E. Cheng, C.-C. Wu, and W.-C. Lee, "Some scheduling problems with deteriorating jobs and learning effects," *Computers & Industrial Engineering*, vol. 54, no. 4, pp. 972–982, 2008.

[4] L. Sun, "Single-machine scheduling problems with deteriorating jobs and learning effects," *Computers and Industrial Engineering*, vol. 57, no. 3, pp. 843–846, 2009.

[5] M. D. Toksarı, "A branch and bound algorithm for minimizing makespan on a single machine with unequal release times under learning effect and deteriorating jobs," *Computers & Operations Research*, vol. 38, no. 9, pp. 1361–1365, 2011.

[6] J.-B. Wang, M.-Z. Wang, and P. Ji, "Single machine total completion time minimization scheduling with a time-dependent learning effect and deteriorating jobs," *International Journal of Systems Science*, vol. 43, no. 5, pp. 861–868, 2012.

[7] J.-B. Wang, L. Liu, and C. Wang, "Single machine SLK/DIF due window assignment problem with learning effect and

deteriorating jobs," *Applied Mathematical Modelling*, vol. 37, no. 18-19, pp. 8394–8400, 2013.

[8] C. Low and W.-Y. Lin, "Some scheduling problems with time-dependent learning effect and deteriorating jobs," *Applied Mathematical Modelling*, vol. 37, no. 20-21, pp. 8865–8875, 2013.

[9] J.-B. Wang and J.-J. Wang, "Flowshop scheduling with a general exponential learning effect," *Computers & Operations Research*, vol. 43, pp. 292–308, 2014.

[10] W.-H. Kuo and D.-L. Yang, "Single-machine group scheduling with a time-dependent learning effect," *Computers & Operations Research*, vol. 33, no. 8, pp. 2099–2112, 2006.

[11] C. C. Wu, Y. R. Shiau, and W. C. Lee, "Single-machine group scheduling problems with deterioration consideration," *Computers and Operations Research*, vol. 35, no. 5, pp. 1652–1659, 2008.

[12] Y. Yan, D.-Z. Wang, D.-W. Wang, W. H. Ip, and H.-F. Wang, "Single machine group scheduling problems with the effects of deterioration and learning," *Acta Automatica Sinica*, vol. 35, no. 10, pp. 1290–1295, 2009.

[13] X. G. Zhang and G. Yan, "Single-machine group scheduling problems with deteriorated and learning effect," *Applied Mathematics & Computation*, vol. 216, no. 4, pp. 1259–1266, 2010.

[14] S.-J. Yang and D.-L. Yang, "Single-machine group scheduling problems under the effects of deterioration and learning," *Computers & Industrial Engineering*, vol. 58, no. 4, pp. 754–758, 2010.

[15] Z. Zhu, L. Sun, F. Chu, and M. Liu, "Single-machine group scheduling with resource allocation and learning effect," *Computers and Industrial Engineering*, vol. 60, no. 1, pp. 148–157, 2011.

[16] W. C. Lee and Z. S. Lu, "Group scheduling with deteriorating jobs to minimize the total weighted number of late jobs," *Applied Mathematics and Computation*, vol. 218, no. 17, pp. 8750–8757, 2012.

[17] X. Huang, M.-Z. Wang, and J.-B. Wang, "Single-machine group scheduling with both learning effects and deteriorating jobs," *Computers & Industrial Engineering*, vol. 60, no. 4, pp. 750–754, 2011.

[18] X. Huang and M.-Z. Wang, "Single machine group scheduling with time and position dependent processing times," *Optimization Letters*, vol. 8, no. 4, pp. 1475–1485, 2014.

[19] M. Ji, K. Chen, J. Ge, and T. C. E. Cheng, "Group scheduling and job-dependent due window assignment based on a common flow allowance," *Computers and Industrial Engineering*, vol. 68, no. 1, pp. 35–41, 2014.

[20] C.-C. Wu and W.-C. Lee, "Single-machine group-scheduling problems with deteriorating setup times and job-processing times," *International Journal of Production Economics*, vol. 115, no. 1, pp. 128–133, 2008.

[21] J.-B. Wang, L. Lin, and F. Shan, "Single-machine group scheduling problems with deteriorating jobs," *The International Journal of Advanced Manufacturing Technology*, vol. 39, no. 7-8, pp. 808–812, 2008.

[22] J. B. Wang, "Single machine scheduling with a time-dependent learning effect and deteriorating jobs," *Journal of the Operational Research Society*, vol. 60, no. 4, pp. 583–586, 2009.

[23] 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.

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

[25] W.-C. Lee and C.-C. Wu, "A note on single-machine group scheduling problems with position-based learning effect," *Applied Mathematical Modelling*, vol. 33, no. 4, pp. 2159–2163, 2009.

[26] Y. Li, G. Li, L. Sun, and Z. Xu, "Single machine scheduling of deteriorating jobs to minimize total absolute differences in completion times," *International Journal of Production Economics*, vol. 118, no. 2, pp. 424–429, 2009.

[27] V. C. Y. Zhu, L. Sun, and X. Li, "Single-machine scheduling time-dependent jobs with resource-dependent ready times," *Computers & Industrial Engineering*, vol. 58, no. 1, pp. 84–87, 2010.

[28] X. Huang, J.-B. Wang, L.-Y. Wang, W.-J. Gao, and X.-R. Wang, "Single machine scheduling with time-dependent deterioration and exponential learning effect," *Computers and Industrial Engineering*, vol. 58, no. 1, pp. 58–63, 2010.

[29] A. Janiak and R. Rudek, "A note on a makespan minimization problem with a multi-ability learning effect," *Omega*, vol. 38, no. 3-4, pp. 213–217, 2010.

[30] M. Ji and T. C. E. Cheng, "Scheduling with job-dependent learning effects and multiple rate-modifying activities," *Information Processing Letters*, vol. 110, no. 11, pp. 460–463, 2010.

[31] S.-J. Yang, "Group scheduling problems with simultaneous considerations of learning and deterioration effects on a single-machine," *Applied Mathematical Modelling*, vol. 35, no. 8, pp. 4008–4016, 2011.

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

[33] J. Bai, Z.-R. Li, and X. Huang, "Single-machine group scheduling with general deterioration and learning effects," *Applied Mathematical Modelling*, vol. 36, no. 3, pp. 1267–1274, 2012.

[34] 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: Systems*, vol. 42, no. 1, pp. 192–200, 2012.

[35] Y. Yin, C.-C. Wu, W.-H. Wu, and J.-C. Chen, "Single-machine group scheduling with a general learning effect," *European Journal of Industrial Engineering*, vol. 7, no. 3, pp. 350–369, 2013.

[36] J.-B. Wang and J.-J. Wang, "Single machine group scheduling with time dependent processing times and ready times," *Information Sciences*, vol. 275, pp. 226–231, 2014.

[37] W. H. Wu, Y. Yin, W. H. Wu, and P. H. Hsu, "A time-dependent scheduling problem to minimize the sum of the total weighted tardiness among two agents," *Journal of Industrial and Management Optimization*, vol. 10, pp. 591–611, 2014.

[38] Y. T. Xu, Y. Zhang, and X. Huang, "Single-machine ready times scheduling with group technology and proportional linear deterioration," *Applied Mathematical Modelling*, vol. 38, no. 1, pp. 384–391, 2014.

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