Research Article

Single-Machine Group Scheduling Problems with Variable Job Processing Times

Ping Ji¹ and Lin Li²

¹Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
²College 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

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 \( j \)-th job in group \( G_i \), \( i = 1, 2, \ldots, m; j = 1, 2, \ldots, n_i \), where \( n_i \) denotes the number of jobs belonging to group \( G_i \) (i.e., \( n_i + n_{i+1} + \cdots + n_m = n \)). Let \( s_j \) be the (actual) setup time of group \( G_i \), and let \( p_{ij} (t) \) 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:

\[
p_{ij} (t) = p_{ij} (A + Bt) f_i (r), \quad (1)
\]

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

\[
s_j = g (u_i), \quad u \leq u_i \leq \bar{u}, \quad i = 1, 2, \ldots, m. \quad (2)
\]
where $u_i$ is the amount of a nonrenewable resource allocated to group $G_i$, with $0 \leq u_i \leq \bar{u}$, $\bar{u}$ is a technological constraint, and $g: R_+ \rightarrow R_+ (R_+ \text{ is the set of nonnegative real numbers})$ is a strictly decreasing continuous function with $g(\bar{u}) \geq 0$. Let $u = (u_1, u_2, \ldots, u_n)$ be the resource allocation vector satisfying the resource constraint $\sum_{i=1}^{n} u_j \leq U$, where $U$ is the total amount of the resource available.

For a given schedule $\pi$, let $C_{ij} = C_{ij}(\pi)$ be the completion time of job $I_{ij}$ under $\pi$, and let $C_{\text{max}} = \max_{i} C_{ij} \mid i = 1, 2, \ldots, m; j = 1, 2, \ldots, n_i$ 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 $1 \mid s_i = g(u_i), p_{ij}(t) = p_{ij}(A \pm Bt) f_i(r), GT, \sum_{j=1}^{n_i} u_j \leq U \mid C_{\text{max}}$ and $1 \mid s_i = g(u_i), p_{ij}(t) = p_{ij}(A \pm Bt) f_i(r), GT, C_{\text{max}} \leq C \mid \sum_{j=1}^{n_i} u_j$.

3. The Problem

**Lemma 1.** For a given schedule $\pi = [I_{11}, I_{12}, \ldots, I_{1n}]$, if the first job starts at time $t_0 \geq 0$, then the completion time $C_{ij}$ is equal to

$$C_{ij} = \left(t_0 + \frac{A}{B}\right) \prod_{i=1}^{j} (1 + Bp_i f_i(0)) = \frac{A}{B}.$$  

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

Proof (by mathematical induction method). For job $I_{11}$, we have

$$C_{11} = t_0 + p_{11} (A \pm Bt_0) f_1(1) = \left(t_0 + \frac{A}{B}\right) (1 + Bp_1 f_1(1)) = \frac{A}{B}. \quad (4)$$

Suppose Lemma 1 holds for job $I_{ij}$; that is,

$$C_{ij} = \left(t_0 + \frac{A}{B}\right) \prod_{i=1}^{j} (1 + Bp_i f_i(i)) = \frac{A}{B}. \quad (5)$$

Then, for job $I_{ij+1}$, we have

$$C_{ij+1} = C_{ij} + p_{ij+1} (A \pm BC_{ij}) f (j + 1) = \left(t_0 + \frac{A}{B}\right) \prod_{i=1}^{j+1} (1 + Bp_i f_i(i)) = \frac{A}{B}. \quad (6)$$

Hence, Lemma 1 holds.

**Theorem 2.** For the $1 \mid s_i = g(u_i), p_{ij}(t) = p_{ij}(A \pm Bt) f_i(r), GT \mid C_{\text{max}}$ problem, and a given schedule $\pi = [G_{11}, G_{12}, \ldots, G_{m1}, J_{11}][1], J_{12}[][2], \ldots, J_{1n}[n_1], J_{21}[][1], J_{22}[][2], \ldots, J_{2n}[n_2], \ldots, J_{m1}[n_1], J_{m2}[n_2], \ldots, J_{mn}[n_n]],$ the makespan is equal to

$$C_{\text{max}} = C_{mn} = \sum_{i=1}^{m} \left(s_i + \frac{A}{B}\prod_{l=1}^{i} (1 + Bp_{il} f_i(l)) \right) \pm \frac{A}{B}. \quad (7)$$

Proof (by mathematical induction method). Stemming from Lemma 1, for group $G_{11}$, we have

$$C_{[1]n_1} = \left(s_{[1]} + \frac{A}{B}\right) \prod_{j=1}^{n_1} (1 + Bp_{11} f_1(j)) \pm \frac{A}{B}. \quad (8)$$

Suppose Theorem 2 holds for group $G_{k1}$, that is,

$$C_{[k]n_k} = \sum_{i=1}^{k} \left(s_{[i]} + \frac{A}{B}\prod_{j=1}^{n_i} (1 + Bp_{i1} f_i(j)) \right) \pm \frac{A}{B}. \quad (9)$$

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

$$C_{[k+1]n_{k+1}} = \left(C_{[k]n_k} + s_{[k+1]} \right) \pm \frac{A}{B} \times \prod_{j=1}^{n_{k+1}} (1 + Bp_{[k+1]1} f_{[k+1]}(j)) \pm \frac{A}{B}$$

$$= \left(\sum_{i=1}^{k} \left(s_{[i]} + \frac{A}{B}\prod_{j=1}^{n_i} (1 + Bp_{i1} f_i(j)) \right) \pm \frac{A}{B} + s_{[k+1]} \right) \pm \frac{A}{B} \times \prod_{j=1}^{n_{k+1}} (1 + Bp_{[k+1]1} f_{[k+1]}(j)) \pm \frac{A}{B}$$

$$= \sum_{i=1}^{k+1} \left(s_{[i]} + \frac{A}{B}\prod_{j=1}^{n_i} (1 + Bp_{i1} f_i(j)) \right) \pm \frac{A}{B} \times \prod_{j=1}^{n_{k+1}} (1 + Bp_{[k+1]1} f_{[k+1]}(j)) \pm \frac{A}{B}$$

$$= \sum_{i=1}^{k+1} \left(s_{[i]} + \frac{A}{B}\prod_{j=1}^{n_i} (1 + Bp_{i1} f_i(j)) \right) \pm \frac{A}{B} \times \prod_{j=1}^{n_{k+1}} (1 + Bp_{[k+1]1} f_{[k+1]}(j)) \pm \frac{A}{B}.$$  

(10)
Theorem 3. For the 1 | \( s_i = g(u_i), p_{ij}(t) = p_{ij}(A \pm Bt) f_i(r) \), \( GT \mid C_{\text{max}} \) 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 | \( s_i = g(u_i), p_{ij}(t) = p_{ij}(A + Bt) f_i(r) \), \( GT \mid C_{\text{max}} \) 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 \( s_{[ij]} \) is bigger than \( \prod_{l=i}^{m} \prod_{j=1}^{n_i} (1 + Bp_{[l][j]} f_{[l]}(j)) \); the makespan can be minimized by the same amount of resource. Obviously

\[
\prod_{l=1}^{m} \prod_{j=1}^{n_i} (1 + Bp_{[l][j]} f_{[l]}(j)) \\
\geq \prod_{l=2}^{m} \prod_{j=1}^{n_i} (1 + Bp_{[l][j]} f_{[l]}(j)) \\
\geq \cdots \geq \prod_{l=m}^{m} \prod_{j=1}^{n_i} (1 + Bp_{[l][j]} f_{[l]}(j)) \geq 1.
\]

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

Theorem 5. For the 1 | \( s_i = g(u_i), p_{ij}(t) = p_{ij}(A - Bt) f_i(r) \), \( GT \mid C_{\text{max}} \) 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 \( s_{[ij]} \) is bigger than \( \prod_{l=i}^{m} \prod_{j=1}^{n_i} (1 - Bp_{[l][j]} f_{[l]}(j)) \); the makespan can be minimized by the same amount of resource. Obviously

\[
\prod_{l=1}^{m} \prod_{j=1}^{n_i} (1 - Bp_{[l][j]} f_{[l]}(j)) \\
\leq \prod_{l=2}^{m} \prod_{j=1}^{n_i} (1 - Bp_{[l][j]} f_{[l]}(j)) \\
\leq \cdots \leq \prod_{l=m}^{m} \prod_{j=1}^{n_i} (1 - Bp_{[l][j]} f_{[l]}(j)) \leq 1.
\]

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

Theorem 6. For the 1 | \( s_i = g(u_i), p_{ij}(t) = p_{ij}(A \pm Bt) f_i(r) \), \( GT \mid C_{\text{max}} \) 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 \( p_{G_i} = \prod_{j=1}^{n_i} (1 \pm Bp_{ij} f_{ij}(j)) \).

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

\[
C_{\text{max}}(\pi) = f(u_{[1]}) \prod_{l=1}^{m} \prod_{j=1}^{n_i} (1 \pm Bp_{[l][j]} f_{[l]}(j)) \\
+ f(u_{[2]}) \prod_{l=2}^{m} \prod_{j=1}^{n_i} (1 \pm Bp_{[l][j]} f_{[l]}(j)) + \cdots \\
+ f(u_{[v-1]}) \prod_{l=v-1}^{m} \prod_{j=1}^{n_i} (1 \pm Bp_{[l][j]} f_{[l]}(j)) \\
+ f(u_{[v]}) \prod_{l=v+2}^{m} \prod_{j=1}^{n_i} (1 \pm Bp_{[l][j]} f_{[l]}(j)) \\
\times \prod_{j=1}^{n_{v+1}} (1 \pm Bp_{[v+1][j]} f_{[v+1]}(j)) \\
+ f(u_{[v+1]}) \prod_{l=v+2}^{m} \prod_{j=1}^{n_i} (1 \pm Bp_{[l][j]} f_{[l]}(j)) + \cdots \\
+ f(u_{[m]}) \prod_{l=m}^{n_i} (1 \pm Bp_{[l][j]} f_{[l]}(j)) \\
\pm \frac{A}{B} \prod_{j=1}^{n_i} (1 \pm Bp_{[l][j]} f_{[l]}(j)) \mp \frac{A}{B},
\]

\[
C_{\text{max}}(\pi') = f(u_{[1]}) \prod_{l=1}^{m} \prod_{j=1}^{n_i} (1 \pm Bp_{[l][j]} f_{[l]}(j)) \\
+ f(u_{[2]}) \prod_{l=2}^{m} \prod_{j=1}^{n_i} (1 \pm Bp_{[l][j]} f_{[l]}(j)) + \cdots \\
+ f(u_{[v-1]}) \prod_{l=v-1}^{m} \prod_{j=1}^{n_i} (1 \pm Bp_{[l][j]} f_{[l]}(j)) \\
+ f(u_{[v+1]}) \prod_{l=v+2}^{m} \prod_{j=1}^{n_i} (1 \pm Bp_{[l][j]} f_{[l]}(j)) \\
+ \cdots \\
+ f(u_{[m]}) \prod_{l=m}^{n_i} (1 \pm Bp_{[l][j]} f_{[l]}(j))
\]
Mathematical Problems in Engineering

\[ + f (u_{[i]}) \prod_{l=1}^{m} \prod_{j=1}^{n_l} (1 + B_{p_{[i][j]},f_{[i]}}(j)) \]
\[ \times \prod_{j=1}^{n_l} (1 + B_{p_{[i]},f_{[i]}}(j)) \]
\[ \times \prod_{l=1}^{m} (1 + B_{p_{[i][j]},f_{[i]}}(j)) \]
\[ + f (u_{[i+1]}) \prod_{l=1}^{m} \prod_{j=1}^{n_l} (1 + B_{p_{[i][j]},f_{[i]}}(j)) \]
\[ + f (u_{[i+2]}) \prod_{l=1}^{m} \prod_{j=1}^{n_l} (1 + B_{p_{[i][j]},f_{[i]}}(j)) + \cdots \]
\[ + f (u_{[n]}) \prod_{l=1}^{m} \prod_{j=1}^{n_l} (1 + B_{p_{[i][j]},f_{[i]}}(j)) \]
\[ \pm \frac{A}{B} \prod_{l=1}^{m} \prod_{j=1}^{n_l} (1 + B_{p_{[i][j]},f_{[i]}}(j)) = \frac{A}{B}. \]

Then

\[ C_{\text{max}}(\pi') - C_{\text{max}}(\pi) \]
\[ = f (u_{[i+1]}) \prod_{l=1}^{m} \prod_{j=1}^{n_l} (1 + B_{p_{[i][j]},f_{[i]}}(j)) \]
\[ \times (\prod_{j=1}^{n_l} (1 + B_{p_{[i+1][j]},f_{[i+1]}}(j)) - \prod_{j=1}^{n_l} (1 + B_{p_{[i][j]},f_{[i]}}(j)) \]
\[ \leq 0 \]
\[ \text{if and only if} \prod_{l=1}^{m} (1 + B_{p_{[i][j]},f_{[i]}}(j)) \leq \prod_{l=1}^{m} (1 + B_{p_{[i][j]},f_{[i]}}(j)). \]

Based on Theorems 3, 4, and 6, an algorithm can be proposed to solve 1 \mid s_i = g(u_i), p_{ij}(r) = p_{ij}(A - Br)f_{ij}(r), GT, \sum_{j=1}^{n} u_{j} \leq U \mid C_{\text{max}}.

Algorithm 7. We have the following.

Step 1. Jobs in each group are scheduled by the SPT rule; that is,
\[ p_{[1]} \leq p_{[2]} \leq \cdots \leq p_{[n]}, \quad i = 1, 2, \ldots, m. \]

Step 2. Groups are scheduled in nonincreasing order of \( \rho_{G_{[i]}} = \prod_{j=1}^{n} (1 + B_{p_{[i][j]},f_{[i]}}(j)) \); that is,
\[ \rho_{G_{[1]}} \geq \rho_{G_{[2]}} \geq \cdots \geq \rho_{G_{[n]}}. \]

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 \( u_{[i]}^* = \min \{l, U\}, U = U - u_{[i]}^* \), 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_{ij}(r) = p_{ij}(A - Br)f_{ij}(r), GT, \sum_{j=1}^{n} u_{j} \leq U \mid C_{\text{max}}.

Algorithm 8. We have the following.

Step 1. Jobs in each group are scheduled by the SPT rule; that is,
\[ p_{[1]} \leq p_{[2]} \leq \cdots \leq p_{[n]}, \quad i = 1, 2, \ldots, m. \]

Step 2. Groups are scheduled in nonincreasing order of \( \rho_{G_{[i]}} = \prod_{j=1}^{n} (1 + B_{p_{[i][j]},f_{[i]}}(j)) \); that is,
\[ \rho_{G_{[1]}} \geq \rho_{G_{[2]}} \geq \cdots \geq \rho_{G_{[n]}}. \]

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 \( u_{[i]}^* = \min \{l, U\}, U = U - u_{[i]}^* \), and \( l = l + 1 \).

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

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, \( A = 1, B = 0.1, U = 5 \), 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 \( J_{12} \rightarrow J_{11} \). In group \( G_2 \), the optimal job sequence is \( J_{22} \rightarrow J_{21} \). In group \( G_3 \), the optimal job sequence is \( J_{31} \rightarrow J_{32} \).

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 \( u_i^* = \min \{U, U\} = 5 \), \( u_2^* = 5, u_1^* = 0 \).
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:

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

Hence \( C_{\text{max}} = C_{12} = 1454.879. \)

4. The Problem

\[
\begin{align*}
  &\text{1 | } s_j = g(u_j), \quad p_{ij}(t) = p_{ij}(A \pm Bt)f_i(r), \quad GT, \quad C_{\text{max}} \leq C \mid \sum u_j
\end{align*}
\]

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 \( \rho_{G_{[m]}} = \prod_{j=1}^{m}(1 \pm Bp_{[ij][j]}f_i(j)) \) and the groups scheduled earlier (later) should be given the priority to resource allocation for the problem \( 1 \mid s_j = g(u_j), p_{ij}(t) = p_{ij}(A + Bt)f_i(r), \) \( GT, \) \( C_{\text{max}} \leq C \mid \sum u_j \).

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

\[
C_{\text{max}} = \sum_{j=1}^{m} \left( f \prod_{l=1}^{m} \prod_{j=1}^{n_i} \left( 1 \pm Bp_{[ij][j]}f_i(j) \right) \right) + \frac{A}{B} \sum_{j=1}^{m} \prod_{j=1}^{n_i} \left( 1 \pm Bp_{[ij][j]}f_i(j) \right) \tag{19}
\]

So, the schedule \( \pi \) is feasible only if

\[
C_{\text{max}} = \sum_{j=1}^{m} \left( f \prod_{l=1}^{m} \prod_{j=1}^{n_i} \left( 1 \pm Bp_{[ij][j]}f_i(j) \right) \right) + \frac{A}{B} \sum_{j=1}^{m} \prod_{j=1}^{n_i} \left( 1 \pm Bp_{[ij][j]}f_i(j) \right) \tag{20}
\]

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, \( C_{\text{max}} = C. \)

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

For the problem \( 1 \mid s_j = g(u_j), p_{ij}(t) = p_{ij}(A + Bt)f_i(r), \) \( GT, \) \( C_{\text{max}} \leq C \mid \sum u_j, \) 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

\[
C = s_{[1]} \prod_{l=1}^{n_1} \left( 1 + Bp_{[ij][j]}f_i(j) \right) + A \sum_{j=1}^{m} \prod_{j=1}^{n_i} \left( 1 + Bp_{[ij][j]}f_i(j) \right) \tag{21}
\]

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

\[
C_{\text{max}} = C - K - s \sum_{l=2}^{m} \prod_{j=1}^{n_j} \left( 1 + Bp_{[ij][j]}f_i(j) \right) - A/B \sum_{j=1}^{m} \prod_{j=1}^{n_i} \left( 1 + Bp_{[ij][j]}f_i(j) \right) + K, 
\]

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

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

\[
\text{If } s_{[1]} < g(\overline{u}), \text{ then the optimal resource allocation of group } G_{[1]} \text{ should be } \overline{u}; \text{ that is, } u_{[1]}^* = \overline{u}. \text{ And the optimal resource consumption } u_{[j]}^*, j = 2, \ldots, n, \text{ can be calculated by the same method.}
\]
Based on the above analysis, we propose an algorithm to solve the problem \( 1 \mid s_i = g(u_i), p_{ij}(t) = p_{ij}(A + Bt) f_i(r), \) \( G T, C_{\text{max}} \leq C \mid \sum u_j. \)

**Algorithm 10.** We have the following.

1. Jobs in each group are scheduled in nondecreasing order of \( p_{ij}; \) that is,
   \[
P_{[1]} \leq P_{[2]} \leq \cdots \leq P_{[m]}, \quad i = 1, 2, \ldots, m.
   \]
2. Groups are scheduled in nonincreasing order of \( \rho_{c_{[i]}} = \prod_{j=1}^{n_i} (1 + B p_{[j][j]} f_i(j)) \); that is,
   \[
   \rho_{c_{[i]}} \geq \rho_{c_{[i-1]}} \geq \cdots \geq \rho_{c_{[m]}}.
   \]
3. For the obtained schedule \( \pi = [G_{[1]}, G_{[2]}, \ldots, G_{[m]}], \) set \( u_{[i]}^* = 0 \) \( (i = 1, 2, \ldots, m), U = 0, s = g(0), s^* = g(\bar{u}), l = 1, \) and \( C' = C - K. \)
4. Calculate
   \[
s_{[i]} = \frac{C' - s \sum_{l=1}^{m} \left( \prod_{k=1}^{m} \prod_{j=1}^{n_i} (1 + B p_{[k][j]} f_i(j)) \right)}{\prod_{l=1}^{m} \prod_{j=1}^{n_i} (1 + B p_{[j][j]} f_i(j))}.
   \]
5. If \( l = m + 1, \) stop; else go to Step 5.
6. **Step 5.** Set \( u_{[i]}^* = \bar{u}, U = U + u_{[i]}^*, C' = C' - s^* \sum_{l=1}^{m} \prod_{j=1}^{n_i} (1 + B p_{[j][j]} f_i(j)), \) and \( l = l + 1. \) If \( l > m, \) then stop; there is no feasible solution; else go to Step 4.

For the problem \( 1 \mid s_i = g(u_i), p_{ij}(t) = p_{ij}(A + Bt) f_i(r), \) \( G T, C_{\text{max}} \leq C \mid \sum u_j, \) 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

\[
C = s \sum_{i=1}^{m-1} \left( \prod_{j=1}^{n_i} (1 - B p_{[i][j]} f_i(j)) \right) + s_{[m]} \left( \prod_{j=1}^{n_m} (1 - B p_{[m][j]} f_m(j)) \right).
\]

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

\[
s_{[m]} = \frac{C - K - s \sum_{l=1}^{m-1} \left( \prod_{j=1}^{n_i} (1 - B p_{[i][j]} f_i(j)) \right)}{\prod_{l=1}^{m} \prod_{j=1}^{n_i} (1 - B p_{[j][j]} f_i(j))}.
\]

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

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

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

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

**Algorithm 11.** We have the following.

1. Jobs in each group are scheduled in nondecreasing order of \( p_{ij}; \) that is,
   \[
P_{[1]} \leq P_{[2]} \leq \cdots \leq P_{[m]}, \quad i = 1, 2, \ldots, m.
   \]
2. Groups are scheduled in nonincreasing order of \( \rho_{c_{[i]}} = \prod_{j=1}^{n_i} (1 + B p_{[j][j]} f_i(j)) \); that is,
   \[
   \rho_{c_{[i]}} \geq \rho_{c_{[i-1]}} \geq \cdots \geq \rho_{c_{[m]}}.
   \]
3. For the obtained schedule \( \pi = [G_{[1]}, G_{[2]}, \ldots, G_{[m]}], \) set \( u_{[i]}^* = 0 \) \( (i = 1, 2, \ldots, m), U = 0, s = g(0), s^* = g(\bar{u}), l = m, \) and \( C' = C - K. \)
4. Calculate
   \[
s_{[i]} = \frac{C' - s \sum_{l=1}^{m} \left( \prod_{k=1}^{m} \prod_{j=1}^{n_i} (1 + B p_{[k][j]} f_i(j)) \right)}{\prod_{l=1}^{m} \prod_{j=1}^{n_i} (1 + B p_{[j][j]} f_i(j))}.
   \]
5. If \( l = 0, \) stop; else go to Step 5.
6. **Step 5.** Set \( u_{[i]}^* = \bar{u}, U = U + u_{[i]}^*, C' = C' - s^* \sum_{l=1}^{m} \prod_{j=1}^{n_i} (1 - B p_{[j][j]} f_i(j)), \) 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 $n = 6, m = 3, f_j(r) = r^m$, 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, $A = 1, B = 0.1, \bar{n} = 10$, 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 $[J_{31} \rightarrow J_{32}] \rightarrow [J_{22} \rightarrow J_{21}] \rightarrow [J_{12} \rightarrow J_{11}]$. Set $u_i^* = 0$ $(i = 1, 2, \ldots, m), U = 0, s = g(0) = 26, s' = g(\bar{n}) = 6, l = 1, K = 1284.072,$ and $C' = C - K = 3000 - 1284.072 = 1715.928$.

Step 4. Calculate

$$s_{[1]} = s_3 = \frac{C' - s \sum_{i=1}^{m} \prod_{i=1}^{m} \prod_{j=1}^{n} (1 + Bp_{i,j}[j]f_{i,j}(j))}{\prod_{i=1}^{m} \prod_{j=1}^{n} (1 + BP_{i,j}[j],f_{i,j}(j))} = 8.6391.$$  

(32)

Steps 5 and 6. Consider $6 < 8.6391 < 26, u_1^* = g^{-1}(8.6391) = 8.6805, U = U + u_1^* = 8.6805, C' = C' - s' \sum_{i=1}^{m} \prod_{j=1}^{n} (1 + BP_{i,j}[j],f_{i,j}(j)) = 1577.935, and l = 2.

Similarly $s_{[2]} = s_2 = 76.72711 > 26; \text{stop. Hence, } u_2^* = u_1^* = 8.6805, u_3^* = u_4^* = 0, \text{and } U = 10.$

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:

$$s_3 = 8.6391,$$

$$C_{31} = 8.6391 + 15 \ast (1 + 0.1 \ast 8.6391) \ast 1^{-0.1} = 36.5978,$$

$$C_{32} = 36.5978 + 18 \ast (1 + 0.1 \ast 36.5978) \ast 2^{-0.1} = 114.8569,$$

$$s_2 = 26,$$

$$C_{22} = 114.8569 + 26 + 10 \ast (1 + 0.1 \ast (114.8569 + 26)) \ast 1^{-0.3} = 291.7138,$$

$$C_{21} = 291.7138 + 20 \ast (1 + 0.1 \ast 291.7138) \ast 2^{-0.3} = 781.8493,$$

$$s_1 = 26,$$

$$C_{12} = 781.8493 + 26 + 8 \ast (1 + 0.1 \ast (781.8493 + 26)) \ast 1^{-0.2} = 1462.129,$$

$$C_{11} = 1462.129 + 12 \ast (1 + 0.1 \ast 1462.129) \ast 2^{-0.2} \approx 3000.$$  

(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 $1 \mid g(u_i), p_{ij}(t) = p_j(A \pm Bt)f_i(r), GT, \sum_{j=1}^{m} u_j \leq U \mid C_{\max}$ and $1 \mid g(u_i), p_{ij}(t) = p_j(A \pm Bt)f_i(r), GT, C_{\max} \leq C \mid \sum u_j$, 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


Submit your manuscripts at http://www.hindawi.com