Research Article

Single-Machine Group Scheduling Problems with Deterioration to Minimize the Sum of Completion Times

Yong He¹ and Li Sun¹, ²

¹ School of Economics and Management, Southeast University, Nanjing 210096, China
² Foundation Department, Huaiyin Advanced Vocational and Technical Health School, Huaian 223300, China

Correspondence should be addressed to Yong He, heyong@126.com

Received 14 August 2012; Accepted 11 November 2012

Academic Editor: Gradimir Milovanovic

Copyright © 2012 Y. He and L. Sun. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We consider two single-machine group scheduling problems with deteriorating group setup and job processing times. That is, the job processing times and group setup times are linearly increasing (or decreasing) functions of their starting times. Jobs in each group have the same deteriorating rate. The objective of scheduling problems is to minimize the sum of completion times. We show that the sum of completion times minimization problems remains polynomially solvable under the agreeable conditions.

1. Introduction

In classical scheduling problems, the scheduling models routinely assume that job processing times are known and fixed throughout the period of job processing. However, this assumption may be unrealistic in many situations that the processing times of jobs may be prolonged due to deterioration or shortened due to learning over time. Scheduling with deteriorating jobs was first independently introduced by J. N. D. Gupta and S. K. Gupta [1] and Browne and Yechiali [2]. Since then, related models of scheduling with deteriorating jobs have been extensively studied from a variety of perspectives. Cheng et al. [3] give a detailed review of scheduling problems with deteriorating jobs. More recent papers which have considered scheduling jobs with deteriorating jobs include Wang [4], Voutsinas and Pappis [5], Lee et al. [6], Wang et al. [7], Zhu et al. [8], Cheng et al. [9, 10], Cheng et al. [11], Sun et al. [12], Wei et al. [13], Zhao and Tang [14], and Yin et al. [15, 16].

On the other hand, the production efficiency can be increased by grouping various parts and products with similar designs and/or production processes. This phenomenon
is known as group technology in the literature. To the best of our knowledge, only a few results concerning scheduling problems with deteriorating effect under group technology are given. Wei and Wang [17] consider single-machine scheduling problems with group technology (GT) and deteriorating jobs. Yang [18] investigate group scheduling problems with simultaneous considerations of learning and deterioration effects on a single-machine setting. The learning phenomenon is implemented to model the setup time of groups. Three models of deteriorating for the job processing time within a group are examined. Bai et al. [19] consider single-machine group scheduling problems with effects of learning and deterioration at the same time. Lee and Lu [20] discuss a single-machine scheduling problem of minimizing the total weighted number of late jobs with deteriorating jobs and setup times. A branch-and-bound with several dominance properties and a lower bound is developed to solve the problem optimally. S. J. Yang and D. L. Yang [21] study two models of learning for the job processing time, they provide polynomial time solutions for the makespan minimization problems. Huang et al. [22] present a single-machine group scheduling problems with both learning effects and deteriorating jobs. Wu and Lee [23] consider a situation where both setup times and job-processing times are lengthened as jobs wait to be processed. Two single-machine group-scheduling problems are investigated where the group setup times and the job-processing times are both increasing functions of their starting times.

In this paper, we study the group scheduling model proposed by Lee and Wu [24], who have considered the makespan problems. In our study, we investigate the sum of completion times minimization problem under the same model as Lee and Wu [24]. In addition, we suppose jobs in each group have the same deteriorating rate. Hence, the problems considered will be noted by $1/GT p_{ij} = a_{ij} + b_j t, s_i = \theta_i + \delta_i t/ \sum C_{ij}$. We show that the problems remain polynomially solvable under the agreeable conditions.

The remainder of this paper is organized in four sections. The solution procedure for the linear increasing function problem is given in the next section, and the solution procedure for the linear decreasing function is described in Section 3. Then, some examples are offered in Section 4. The conclusion is given in the last section.

2. The $1/GT, p_{ij} = a_{ij} + b_j t, s_i = \theta_i + \delta_i t/ \sum C_{ij}$ Problem

In this section, we consider the single-machine group scheduling problem with deterioration to minimize the sum of completion times of all jobs. In our model, we follow the notation and terminology used by Lee and Wu [24]. In addition, $a_{i[j]}$ is the basic (normal) processing time of job $J_{i[j]}$ scheduled in the $j$th position of group $G_i$, and $C_{i[j]}$ is the completion time scheduled in the $j$th position of group $G_i$. We suppose the jobs in each group have the same deterioration rate noted by $b_i$.

**Theorem 2.1.** For the $1/GT, p_{ij} = a_{ij} + b_j t, s_i = \theta_i + \delta_i t/ \sum C_{ij}$ problem, the optimal schedule is obtained by sequencing the jobs in each group in nondecreasing order of $a_{i[j]}$, that is, $a_{i[1]} \leq a_{i[2]} \leq a_{i[3]} \leq a_{i[4]} \leq \cdots \leq a_{i[m]}, i = 1, 2, 3, \ldots m$, (the SPT rule).

**Proof.** In the same group, the result can be easily obtained by using interchanging technology and is omitted. $\square$

**Theorem 2.2.** For the $1/GT, p_{ij} = a_{ij} + b_j t, s_i = \theta_i + \delta_i t/ \sum C_{ij}$ problem, the groups are arranged in nondecreasing order of $\omega(G_i), \omega(G_i) = b_i ((1 + b_i)^n - 1) / (1 + b_i)((1 + b_i)^n - 1)(1 + \delta_i)$,
if \( \omega(G_i) \) and \( B_i/A_i \) have agreeable conditions, that is, \( \omega(G_i) \geq \omega(G_j) \) if and only if \( B_i/A_i \geq B_j/A_j \),

\[ A_i = (1 + b_i)(1 - (1 + b_i)^{n_i})(1 + \delta_i)/(1 - (1 + b_i)), \quad B_i = \sum_{k=1}^{n_i} (1 + b_i)^{n_i-k} a_i[k] + (1 + b_i)^{n_i}\theta_i, \]

**Proof.** Let \( \pi_1 \) and \( \pi_2 \) be two job schedules where the difference between \( \pi_1 \) and \( \pi_2 \) is a pairwise interchange of two adjacent groups \( G_i \) and \( G_j \). That is, \( \pi_1 = [S_1, G_i, G_j, S_2] \), \( \pi_2 = [S_1, G_i, G_j, S_2] \), where \( S_1 \) and \( S_2 \) are partial sequences and \( S_1 \) and \( S_2 \) may be empty. Furthermore, we assume that \( t \) denotes the completion time of the last job in \( \pi_1 \). To show \( \pi_1 \) dominates \( \pi_2 \), it suffices to show that

\[
\sum_{i=1}^{n_1} C_{i[i]}(\pi_1) + \sum_{i=1}^{n_1} C_{j[i]}(\pi_1) \leq \sum_{i=1}^{n_1} C_{j[i]}(\pi_2) + \sum_{i=1}^{n_1} C_{i[i]}(\pi_2).
\]

By definition, the completion times of jobs in \( \pi_1 \) are given by

\[
C_{i[i]}(\pi_1) = a_{i[i]} + (1 + b_i)\theta_i + (1 + b_i)(1 + \delta_i)t,
\]

\[
C_{j[i]}(\pi_1) = a_{j[i]} + (1 + b_j)\theta_j + (1 + b_j)(1 + \delta_j)C_{i[i]}(\pi_1),
\]

\[
C_{j[i]}(\pi_2) = a_{j[i]} + (1 + b_j)\theta_j + (1 + b_j)^n(1 + \delta_j)C_{i[i]}(\pi_2).
\]

And the completion times of jobs in \( \pi_2 \) are given by

\[
C_{i[i]}(\pi_2) = a_{i[i]} + (1 + b_i)\theta_i + (1 + b_i)(1 + \delta_i)t,
\]

\[
C_{j[i]}(\pi_2) = a_{j[i]} + (1 + b_i)\theta_i + (1 + b_i)(1 + \delta_i)C_{j[i]}(\pi_2).
\]
Therefore, we have

\[
\sum_{i=1}^{n_i} C_{ij[i]}(\pi_2) + \sum_{i=1}^{n_i} C_{ij[i]}(\pi_2) - \sum_{i=1}^{n_i} C_{ij[i]}(\pi_1) - \sum_{i=1}^{n_i} C_{ij[i]}(\pi_1) \\
= \frac{(1 + b_j)(1 - (1 + b_1)^{n_j})(1 + \delta_j)t}{1 - (1 + b_j)} + \frac{(1 + b_j)(1 - (1 + b_1)^{n_j})(1 + \delta_j)C_{j[n_j]}(\pi_2)}{1 - (1 + b_j)} \\
- \frac{(1 + b_j)(1 - (1 + b_1)^{n_j})(1 + \delta_j)t}{1 - (1 + b_j)} - \frac{(1 + b_j)(1 - (1 + b_1)^{n_j})(1 + \delta_j)C_{j[n_j]}(\pi_1)}{1 - (1 + b_j)}. \\
\]

(2.4)

By substituting

\[
A_j = \frac{(1 + b_j)(1 - (1 + b_1)^{n_j})(1 + \delta_j)}{1 - (1 + b_j)}, \quad A_j = \frac{(1 + b_j)(1 - (1 + b_1)^{n_j})(1 + \delta_j)}{1 - (1 + b_j)}, \\
B_i = \sum_{k=1}^{n_i} (1 + b_1)^{n_i-k} a_{i[k]} + (1 + b_1)^{n_i} \theta_i, \quad B_j = \sum_{k=1}^{n_j} (1 + b_j)^{n_j-k} a_{j[k]} + (1 + b_j)^{n_j} \theta_j,
\]

we can obtain

\[
\sum_{i=1}^{n_i} C_{j[i]}(\pi_2) + \sum_{i=1}^{n_i} C_{j[i]}(\pi_2) - \sum_{i=1}^{n_i} C_{j[i]}(\pi_1) - \sum_{i=1}^{n_i} C_{j[i]}(\pi_1) \\
= (A_i((1 + b_j)^{n_j}(1 + \delta_j) - 1) - A_j((1 + b_j)^{n_j}(1 + \delta_i) - 1))t + A_iB_j - A_jB_i.
\]

(2.6)

To have

\[
\sum_{i=1}^{n_i} C_{j[i]}(\pi_2) + \sum_{i=1}^{n_i} C_{j[i]}(\pi_2) - \sum_{i=1}^{n_i} C_{j[i]}(\pi_1) - \sum_{i=1}^{n_i} C_{j[i]}(\pi_1) \geq 0,
\]

(2.7)

when \( t \geq 0 \), if and only if

\[
A_i((1 + b_j)^{n_j}(1 + \delta_j) - 1) - A_j((1 + b_j)^{n_j}(1 + \delta_i) - 1) \geq 0, \quad A_iB_j - A_jB_i \geq 0.
\]

(2.8)

(2.9)

From the result of (2.8), we have \(((1 + b_j)^{n_j}(1 + \delta_j) - 1)/A_i \leq ((1 + b_j)^{n_j}(1 + \delta_j) - 1)/A_j\), that is to say \( \omega(G_i) \leq \omega(G_j) \).

And from the case of (2.9), we can obtain \( B_i/A_i \leq B_j/A_j \).

Therefore, if \( \omega(G_i) \) and \( B_i/A_i \) have agreeable conditions, the optimal sequence between groups are arranged in nondecreasing order of \( \omega(G_i) \), This completes the proof. \( \square \)

From Theorems 2.1 and 2.2, if \( \omega(G_i) \) and \( B_i/A_i \) have agreeable condition, the problem \( 1/GT, p_{ij} = a_{ij} + b_it, s_i = \theta_i + \delta_i t/\sum C_{ij} \) can be solved by the following algorithm.
**Theorem 3.2.** For the\( \omega \) rule, Jobs in each group are scheduled in nondecreasing order of the basic processing time \( a_{ij} \), that is, \( a_{i[1]} \leq a_{i[2]} \leq a_{i[3]} \leq \cdots \leq a_{i[n_i]} \), \( i = 1, 2, 3, \ldots m \), (the SPT rule).

Step 2. Calculate \( \omega(G_i) = b_i((1 + b_i)^n - 1)/(1 + b_i)^n - 1)(1 + \delta_i) \).

Step 3. Groups are scheduled in nondecreasing order of \( \omega(G_i) \), that is, \( \omega(G_1) \leq \omega(G_2) \leq \cdots \leq \omega(G_m) \).

Obviously, it is easy to show that the total time for Algorithm 2.3 is \( O(n \log n) \).

**3. The 1/GT, \( p_{ij} = a_{ij} - b_i t \), \( s_i = \theta_i - \delta_i t \)/\( \sum C_{ij} \) Problem**

In order to discuss the problem 1/GT, \( p_{ij} = a_{ij} - b_i t \), \( s_i = \theta_i - \delta_i t \)/\( \sum C_{ij} \) conveniently, we suppose \( 0 \leq b_i \leq 1 \) and \( 0 \leq \theta_i \leq 1 \).

**Theorem 3.1.** For the 1/GT, \( p_{ij} = a_{ij} - b_i t \), \( s_i = \theta_i - \delta_i t \)/\( \sum C_{ij} \) problem, the optimal schedule is obtained by sequencing the jobs in each group in nondecreasing order of basic processing time \( a_{ij} \), that is, \( a_{i[1]} \leq a_{i[2]} \leq a_{i[3]} \leq \cdots \leq a_{i[n_i]} \), \( i = 1, 2, 3, \ldots m \), (the SPT rule).

**Proof.** In the same group, the result can be easily obtained by using interchanging technology and is omitted.

**Theorem 3.2.** For the 1/GT, \( p_{ij} = a_{ij} - b_i t \), \( s_i = \theta_i - \delta_i t \)/\( \sum C_{ij} \) problem, the groups are arranged in nondecreasing order of \( \omega(G_i) \), \( \omega(G_i) = b_i((1 - b_i)^n - 1)/(1 - b_i)^n - 1)(1 - \delta_i) \), if \( \omega(G_i) \) and \( B_i/A_i \) have agreeable conditions, that is, \( B_i/A_i \geq B_j/A_j \) if and only if \( \omega(G_i) \geq \omega(G_j) \), \( A_i = (1 - b_i)(1 - (1 - b_i)^n)(1 - \delta_i) / (1 - b_i) \), and \( B_i = \sum_{k=1}^{n_i}(1 - b_i)^{n_i-k}a_{i[k]} + (1 - b_i)^{n_i}\theta_i \).

**Proof.** Here, we still use the same notations as that in Theorem 2.2. To show \( \pi_1 \) dominates \( \pi_2 \), it suffices to show that

\[
\sum_{i=1}^{n_i} C_{i[1]}(\pi_1) + \sum_{j=1}^{n_j} C_{j[j]}(\pi_1) \leq \sum_{i=1}^{n_i} C_{i[1]}(\pi_2) + \sum_{j=1}^{n_j} C_{j[j]}(\pi_2).
\]

By definition, the completion times of jobs in \( \pi_1 \) are given by

\[
C_{i[1]}(\pi_1) = a_{i[1]} + (1 - b_i)\theta_i + (1 - b_i)(1 - \delta_i)t,
\]

\[
C_{i[n_i]}(\pi_1) = \sum_{k=1}^{n_i}(1 - b_i)^{n_i-k}a_{i[k]} + (1 - b_i)^{n_i}\theta_i + (1 - b_i)^{n_i}(1 - \delta_i)t,
\]

\[
C_{j[1]}(\pi_1) = a_{j[1]} + (1 - b_j)\theta_j + (1 - b_j)(1 - \delta_j)C_{i[n_i]}(\pi_1),
\]

\[
C_{j[n_j]}(\pi_1) = \sum_{k=1}^{n_j}(1 - b_j)^{n_j-k}a_{j[k]} + (1 - b_j)^{n_j}\theta_j + (1 - b_j)^{n_j}(1 - \delta_j)C_{i[n_i]}(\pi_1).
\]
And the completion times of jobs in $\tau_2$ are given by

$$C_{i[1]}(\tau_2) = a_{i[1]} + (1 - b_i)\theta_j + (1 - b_i)(1 - \delta_i)t,$$

$$\vdots$$

$$C_{i[n_i]}(\tau_2) = \sum_{k=1}^{n_i} (1 - b_i)^{n_i-k} a_{i[k]} + (1 - b_i)^{n_i} \theta_j + (1 - b_i)^{n_i}(1 - \delta_i)t,$$

$$C_{i[1]}(\tau_2) = a_{i[1]} + (1 - b_i)\theta_j + (1 - b_i)(1 - \delta_i)C_{i[n_i]}(\tau_2),$$

$$\vdots$$

$$C_{i[n_i]}(\tau_2) = \sum_{k=1}^{n_i} (1 - b_i)^{n_i-k} a_{i[k]} + (1 - b_i)^{n_i} \theta_j + (1 - b_i)^{n_i}(1 - \delta_i)C_{i[n_i]}(\tau_2).$$

Therefore, we have

$$\sum_{i=1}^{n_i} C_{i[i]}(\tau_2) + \sum_{i=1}^{n_i} C_{i[i]}(\tau_2) - \sum_{i=1}^{n_i} C_{i[i]}(\tau_1) - \sum_{i=1}^{n_i} C_{i[i]}(\tau_1)$$

$$= \frac{(1 - b_i)(1 - (1 - b_i)^{n_i})(1 - \delta_i)t}{1 - (1 - b_i)} + \frac{(1 - b_i)(1 - (1 - b_i)^{n_i})(1 - \delta_i)C_{i[n_i]}(\tau_2)}{1 - (1 - b_i)}$$

$$- \frac{(1 - b_i)(1 - (1 - b_i)^{n_i})(1 - \delta_i)t}{1 - (1 - b_i)} - \frac{(1 - b_i)(1 - (1 - b_i)^{n_i})(1 - \delta_i)C_{i[n_i]}(\tau_1)}{1 - (1 - b_i)}.$$  

(3.4)

By substituting

$$A_i = \frac{(1 - b_i)(1 - (1 - b_i)^{n_i})(1 - \delta_i)}{1 - (1 - b_i)}, \quad A_j = \frac{(1 - b_i)(1 - (1 - b_i)^{n_i})(1 - \delta_i)}{1 - (1 - b_i)},$$

$$B_i = \sum_{k=1}^{n_i} (1 - b_i)^{n_i-k} a_{i[k]} + (1 - b_i)^{n_i} \theta_j, \quad B_j = \sum_{k=1}^{n_i} (1 - b_i)^{n_i-k} a_{i[k]} + (1 - b_i)^{n_i} \theta_j,$$

we can obtain

$$\sum_{i=1}^{n_i} C_{i[i]}(\tau_2) + \sum_{i=1}^{n_i} C_{i[i]}(\tau_2) - \sum_{i=1}^{n_i} C_{i[i]}(\tau_1) - \sum_{i=1}^{n_i} C_{i[i]}(\tau_1)$$

$$= (A_i((1 - b_i)^{n_i}(1 - \delta_i) - 1) - A_j((1 - b_i)^{n_i}(1 - \delta_i) - 1))t + A_iB_j - A_jB_i.$$  

(3.6)

To have

$$\sum_{i=1}^{n_i} C_{i[i]}(\tau_2) + \sum_{i=1}^{n_i} C_{i[i]}(\tau_2) - \sum_{i=1}^{n_i} C_{i[i]}(\tau_1) - \sum_{i=1}^{n_i} C_{i[i]}(\tau_1) \geq 0,$$

(3.7)
when $t \geq 0$ if and only if

$$A_i((1 - b_i)^n(1 - \delta_i) - 1) - A_j((1 - b_i)^n(1 - \delta_i) - 1) \geq 0, \quad (3.8)$$

$$A_iB_j - A_iB_i \geq 0. \quad (3.9)$$

From the result of (3.8), we have

$$\frac{(1 - b_i)^n(1 - \delta_i) - 1}{A_i} \leq \frac{(1 - b_j)^n(1 - \delta_j) - 1}{A_j}, \quad (3.10)$$

that is to say $\omega(G_i) \leq \omega(G_j)$.

And from the case of (3.9), we can obtain $B_i/A_i \leq B_j/A_j$.

Therefore, if $\omega(G_i)$ and $B_i/A_i$ have agreeable conditions, the optimal sequence between groups are arranged in nondecreasing order of $\omega(G_i)$. This completes the proof. \(\square\)

From Theorems 3.1 and 3.2, if $\omega(G_i)$ and $B_i/A_i$ have agreeable condition, the problem $1/\text{GT}, p_{ij} = a_{ij} + b_it, s_i = \theta_i - \delta_it/\sum C_{ij}$ can be solved by the following algorithm.

**Algorithm 3.3.** Step 1. Jobs in each group are scheduled in nondecreasing order of the basic processing time $a_{i[j]}$, that is, $a_{i[1]} \leq a_{i[2]} \leq a_{i[3]} \leq \cdots \leq a_{i[m]}, i = 1, 2, 3, \ldots, m$ (the SPT rule).

Step 2. Calculate $\omega(G_i) = b_i((1 - b_i)^n(1 - \delta_i) - 1)/(1 - b_i)(1 - (1 - b_i)^n)(1 - \delta_i)$.

Step 3. Groups are scheduled in nondecreasing order of $\omega(G_i)$, that is, $\omega(G_1) \leq \omega(G_2) \leq \omega(G_3) \leq \cdots \leq \omega(G_m)$.

Obviously, It is easy to show that the total time for Algorithm 3.3 is $O(n \log n)$.

### 4. Examples

**Example 4.1.** For the $1/\text{GT}, p_{ij} = a_{ij} + b_it, s_i = \theta_i + \delta_it/\sum C_{ij}$ problem, we consider six jobs are divided into three groups. The basic job and group processing times, the job and group deterioration rates for each group are given, respectively, as follows:

$$p_{11} = 5, \quad p_{12} = 8, \quad p_{21} = 4, \quad p_{22} = 10, \quad p_{23} = 6, \quad p_{31} = 7,$$

$$b_1 = 0.04, \quad b_2 = 0.02, \quad b_3 = 0.01, \quad \theta_1 = 5, \quad \theta_2 = 4, \quad \theta_3 = 6, \quad \delta_1 = 0.01, \quad \delta_2 = 0.05, \quad \delta_3 = 0.04. \quad (4.1)$$

**Solution 1.** According to Algorithm 2.3, we solve the example as follows.

**Step 1.** In group $G_1$, the optimal job sequence is $J_{11} \rightarrow J_{12}$.

In group $G_2$, the optimal job sequence is $J_{21} \rightarrow J_{23} \rightarrow J_{22}$.

In group $G_3$, the optimal job sequence is $J_{31}$.

**Step 2.** calculate: $B_1/A_1 = 8.683, B_2/A_2 = 7.483, B_3/A_3 = 12.483$, and $\omega(G_1) = 0.043, \omega(G_2) = 0.035, \omega(G_3) = 0.048$. It is easy to say $B_2/A_2 < B_1/A_1 < B_3/A_3$ and $\omega(G_2) < \omega(G_1) < \omega(G_3)$. 


Therefore, the optimal group sequence is \( G_2 \rightarrow G_1 \rightarrow G_3 \) and the optimal schedule is \([J_{21} \rightarrow J_{23} \rightarrow J_{22}] \rightarrow [J_{11} \rightarrow J_{12}] \rightarrow [J_{31}] \). The completion times of jobs are \( C_{21} = 8.08, C_{23} = 14.242, C_{22} = 24.527, C_{11} = 35.963, C_{12} = 45.402, \) and \( C_{31} = 60.750 \), and the sum of completion times is \( \sum C_{ij} = 8.08 + 14.242 + 24.527 + 35.963 + 45.402 + 60.750 = 188.964 \).

**Example 4.2.** For the \( 1/\text{GT} \), \( p_{ij} = a_{ij} - b_i t, s_i = \theta_i - \delta_i t / \sum C_{ij} \) problem, we consider six jobs are divided into three groups as Example 4.1, the job and group deterioration rates for each group are given, respectively, as follows:

\[
\begin{align*}
  p_{11} &= 9, & p_{12} &= 7, & p_{13} &= 5, & p_{21} &= 10, & p_{31} &= 8, & p_{32} &= 6, \\
  b_1 &= 0.03, & b_2 &= 0.05, & b_3 &= 0.02, & \theta_1 &= 4, & \theta_2 &= 8, & \theta_3 &= 5, \\
  \delta_1 &= 0.15, & \delta_2 &= 0.01, & \delta_3 &= 0.1. 
\end{align*}
\]

**Solution 2.** According to Algorithm 2.3, we solve the example as follows.

**Step 1.** In group \( G_1 \), the optimal job sequence is \( J_{13} \rightarrow J_{12} \rightarrow J_{11} \).
In group \( G_2 \), the optimal job sequence is \( J_{21} \).
In group \( G_3 \), the optimal job sequence is \( J_{32} \rightarrow J_{31} \).

**Step 2.** calculate: \( B_1/A_1 = 10.060, B_2/A_2 = 18.713, B_3/A_3 = 10.700, \) and \( \omega(G_1) = -0.093, \omega(G_2) = -0.063, \omega(G_3) = -0.078 \). It is easy to say \( B_1/A_1 < B_2/A_3 < B_2/A_2 \) and \( \omega(G_1) < \omega(G_3) < \omega(G_2) \). Therefore, the optimal group sequence is \( G_1 \rightarrow G_3 \rightarrow G_2 \) and the optimal schedule is \([J_{13} \rightarrow J_{12} \rightarrow J_{11}] \rightarrow [J_{32} \rightarrow J_{31}] \rightarrow [J_{21}] \). The completion times of jobs are \( C_{13} = 8.88, C_{12} = 15.614, C_{11} = 24.146, C_{32} = 34.326, C_{31} = 41.639, C_{21} = 53.201, \) and the sum of completion times is \( \sum C_{ij} = 8.88 + 15.614 + 24.146 + 34.326 + 41.639 + 53.201 = 177.806 \).

**5. Conclusions**

In this paper, we investigate the single-machine group scheduling problems with deterioration which is studied by Lee and Wu [24]. We consider the sum of completion times minimization problem, and we show the problem to minimize total completion time of all jobs is polynomially solvable under agreeable conditions in our study.

**Acknowledgments**

The authors thank the valuable comments of the referees for an earlier version of this paper. Their comments have significantly improved the paper. This work is supported by the National Natural Science Foundation of China (no. 71001025). Also, this research is partly supported by the Program for New Century Excellent Talents in University (no. NCET-10-0327) and the Ministry of Education of China: Grant-in-aid for Humanity and Social Science Research (no. 11YJCZH139).
References


