Single-machine group scheduling problems with
deterioration consideration

Chin-Chia Wu\textsuperscript{a}, Yau-Ren Shiau\textsuperscript{b}, Wen-Chiung Lee\textsuperscript{a,*}

\textsuperscript{a}Department of Statistics, Feng Chia University, Taichung, Taiwan
\textsuperscript{b}Department of Industrial Engineering and System Management, Feng Chia University, Taichung, Taiwan

Available online 11 October 2006

Abstract

In many realistic situations, a job processed later consumes more time than the same job when it is processed earlier; this phenomenon is known as deteriorating jobs. However, job deterioration is relatively unexplored in the context of group technology. In this paper, we consider deterioration in the framework of minimizing the makespan and the total completion time in single-machine group scheduling problems. We show that these problems remain polynomially solvable when deterioration is considered.

© 2006 Elsevier Ltd. All rights reserved.

Keywords: Group technology; Single machine; Deteriorating jobs; Makespan; Total completion time

1. Introduction

In classical scheduling, most research assumes that processing times of jobs are constant over the entire planning horizon. However, empirical studies have found that, in many scheduling environments, a job that is processed later consumes more time than the same job when processed earlier. For example, the temperature of an ingot, while waiting to enter the rolling machine, drops below a certain level, requiring the ingot to be reheated before rolling. The time and effort required to control a fire increases if there is a delay in the start of the fire-fighting effort. More extensive medical treatment might be necessary as a patient’s health condition worsens [1–3]. Scheduling in this setting is known as scheduling deteriorating jobs.

The deteriorating job scheduling problem was first introduced independently by Gupta and Gupta [1] and Browne and Yechiali [4]. Since then, related models of time-dependent processing times have been extensively studied from a variety of perspectives [5]. Mosheiov [6] presented a problem of minimizing the total completion time where jobs have the same basic processing time and possibly different deteriorating rates. He showed that the optimal sequence to minimize flow time is V-shaped. Cheng and Ding [7] considered a family of scheduling problems for a set of starting time-dependent tasks with release times and linearly increasing or decreasing processing rates on a single machine to minimize the makespan. Ng et al. [8] investigated three scheduling problems with different linearly decreasing functions to minimize the total completion time on a single machine. Cheng and Ding [9] studied the feasibility problem of scheduling a set of start-time-dependent tasks on a single machine with known deadlines and processing rates and identical initial processing times. They showed that the cases with arbitrary deadlines are strongly NP-complete and that the cases...
with two distinct deadlines are NP-complete in the ordinary sense. Wang and Xia [10] addressed no-wait or no-idle flowshop scheduling problems under a simple linear deteriorating function. They showed that the makespan and the total weighted completion time problems remain polynomial solvable, although the solutions are more complicated than the classical ones.

Sundararaghavan and Kunnathur [11] considered a case that certain maintenance procedures might fail to complete prior to a prespecified deadline and need extra time for successful accomplishment. They proposed optimal and heuristic algorithms to minimize the makespan and the total weighted completion time, respectively. Cheng and Ding [12] considered a model where each task has a normal processing time that deteriorates as a step function if its starting time is beyond a given deteriorating date. They focused on problems with identical task deteriorating dates and showed that the flow time problem is NP-complete. Cheng et al. [13] studied scheduling problems for a set of non-preemptive jobs where the processing time of a job is a piecewise non-increasing function of its starting time. They showed that the makespan and the total completion time problems are NP-hard. In addition, Browne and Yechiali [4] introduced the makespan problems with exponential job processing times. Recently, Voutsinas and Pappis [14] introduced a new type of model where the job value deteriorates exponentially over time. Comprehensive reviews of different models and problems concerning jobs with starting time-dependent processing times were given by Alidaee and Womer [5] and Cheng et al. [15].

However, most studies ignore the fact that 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 [16]. Many advantages have been claimed through the wide applications of group technology. For instance, changeover between different parts are simplified, thereby reducing the costs involved; parts spend less time waiting, which results in less work-in-process inventory; parts tend to move through production in a direct route, and hence the manufacturing lead time is reduced; the variability of tasks is reduced, and hence worker training is simplified [17–22]. Based on the fact that the more time spent practicing, the greater the learning effect, Kuo and Yang [23] proposed time-dependent learning effect models where the decrement of a job processing time depends on the total amount of job processing times before it. They brought the concept into single-machine group scheduling problems and showed that the makespan and the total completion time problems remain polynomial solvable. Guo and Wang [24] investigated the makespan minimization problem where the actual processing time is given by

\[ p_{ij}(t) = p_{ij}(a + bt) \]

They showed that the problem is polynomial solvable under the group technology assumption. Under the same model, Xu et al. [25] proved that the total weighted completion time minimization problem remains polynomial solvable. Moreover, Wang et al. [26] showed that single-machine group scheduling problems are polynomially solvable where the objectives are to minimize the makespan and the total completion time under the model that

\[ p_{ij}(t) = a_{ij} - b_{ij}t. \]

However, research on starting time-dependent processing times under group technology assumes that the setup times are constant. Since longer setup or preparation might be necessary as food quality deteriorates or a patient’s condition worsens, we consider a situation where the setup time grows and jobs deteriorate as they wait for processing. To the best of our knowledge, group technology with starting time-dependent setup times has never before been discussed in the context of deteriorating jobs.

In this paper, we consider 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 are the makespan and the total completion time. The remainder of this paper is organized in four sections. Notation is defined and the problem stated in Section 2. The solution procedures for the makespan minimization problem and the total completion time minimization problem are described in Sections 3 and 4, respectively. The conclusion is given in the last section.

2. Problem statement and notation

There are \( n \) jobs classified into \( m \) groups ready to be processed on a single machine. All jobs are available at time \( t_0 \), where \( t_0 > 0 \). Jobs in the same group are processed consecutively. A setup time precedes the processing of each group. The actual job processing time of job \( J_{ij} \) is a simple linear function of its starting time \( t \) [27], that is,

\[ p_{ij} = a_{ij}t, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n_i. \]

Moreover, the actual setup time of group \( G_i \) is also a simple linear function of its starting time \( t \) as follows:

\[ s_i = \delta_i t, \quad i = 1, 2, \ldots, m. \]
Notation used throughout the paper is described next. Square brackets [ ] are used to signify the position of jobs in a sequence.

**Notation:**

- \( m \) the number of groups \( (m \geq 2) \)
- \( G_i \) group \( i, i = 1, 2, \ldots, m \)
- \( G_{i[j]} \) the \( i \)th group in a group sequence, \( i = 1, 2, \ldots, m \)
- \( n_i \) the number of jobs in group \( G_i, i = 1, 2, \ldots, m \)
- \( n_{i[j]} \) the number of jobs in the \( i \)th group in a group sequence, \( i = 1, 2, \ldots, m \)
- \( n \) the total number of jobs (i.e., \( n_1 + n_2 + \cdots + n_m = n \))
- \( J_{ij} \) job \( j \) in group \( G_i, i = 1, 2, \ldots, m, j = 1, 2, \ldots, n_i \)
- \( \delta_i \) the deteriorating rate of the setup time for group \( G_i, i = 1, 2, \ldots, m \)
- \( \delta_{i[j]} \) the deteriorating rate of the setup time for the \( i \)th group in a group sequence, \( i = 1, 2, \ldots, m \)
- \( \alpha_{ij} \) the deteriorating rate of the processing time for the \( j \)th job in group \( G_i \)
- \( \alpha_{[i,j]} \) the deteriorating rate of the processing time for the job scheduled in the \( j \)th position in group \( G_{i[j]} \)
- \( \alpha_{(i,j)} \) the deteriorating rate of the processing time for the \( j \)th job in group \( G_i \) when jobs are arranged in non-decreasing order of their deteriorating rates, that is, \( \alpha_{(1)} \leq \alpha_{(2)} \leq \cdots \leq \alpha_{(n_i)}, i = 1, 2, \ldots, m, j = 1, 2, \ldots, n_i \)
- \( s_i \) the actual setup time of group \( G_i \)
- \( p_{ij} \) the actual processing time of \( J_{ij} \)
- \( S \) a schedule of \( n \) jobs
- \( C_{ij}(S) \) the completion time of \( J_{ij} \) under schedule \( S \)
- \( C_{[k]}(S) \) the completion time of the job scheduled in the \( k \)th position and in group \( G_i \) under schedule \( S \)
- \( C_{[i,k]}(S) \) the completion time of the job scheduled in the \( k \)th position and in the \( i \)th group under schedule \( S \)
- \( C_{\text{max}} \) the makespan of all jobs
- \( \sum C_{ij} \) the total completion of all jobs

Let \( G \) indicate that the problem is a group scheduling problem and \( s_i = \delta_i t \) indicate starting time-dependent setup times. Then, using conventional notation, the makespan and total completion time minimization problems considered in this paper are denoted as \( 1/G, s_i = \delta_i t, p_{ij} = \alpha_{ij} t / C_{\text{max}} \) and \( 1/G, s_i = \delta_i t, p_{ij} = \alpha_{ij} t / \sum C_{ij} \).

### 3. Makespan minimization

In this section, a single-machine group scheduling problem with deterioration consideration is studied. It is assumed that the actual job processing time is a simple linear function of its starting time. Moreover, the actual setup time is also a simple linear function of its starting time. The objective is to minimize the makespan of all jobs. The following theorem provides the optimal schedule for the group sequence and job sequence.

**Theorem 1.** For the \( 1/G, s_i = \delta_i t, p_{ij} = \alpha_{ij} t / C_{\text{max}} \) problem, the optimal schedule satisfies the property that the group sequence and the job sequence in each group can be in any order.

**Proof.** The completion times for jobs in the first group \( G_{[1]} \) under \( S \) are

\[
C_{[1],[1]}(S) = t_0 + \delta_{[1]} t_0 + \alpha_{[1],[1]}(t_0 + \delta_{[1]} t_0) \\
= t_0(1 + \delta_{[1]})(1 + \alpha_{[1],[1]}),
\]

\[
C_{[1],[2]}(S) = C_{[1],[1]}(S) + \alpha_{[1],[2]} C_{[1],[1]}(S) \\
= t_0(1 + \delta_{[1]})(1 + \alpha_{[1],[1]})(1 + \alpha_{[1],[2]}),
\]

\[
: \\
C_{[1],[n_{[1]}]}(S) = t_0(1 + \delta_{[1]}) \prod_{j=1}^{n_{[1]}} (1 + \alpha_{[1],[j]}), \tag{1}
\]
The completion times for jobs in the second group $G_{[2]}$ under $S$ are

$$\begin{align*}
C_{[2],[1]}(S) &= (1 + \delta_{[2]})C_{[1],n_{[1]}}(S) + \alpha_{[2],[1]}(1 + \delta_{[2]})C_{[1],n_{[1]}}(S) \\
&= t_0(1 + \delta_{[1]})(1 + \delta_{[2]}) \prod_{j=1}^{n_{[1]}} (1 + \alpha_{[1],j})(1 + \alpha_{[2],[1]}),
\end{align*}$$

\[C_{[2],[2]}(S) = t_0(1 + \delta_{[1]})(1 + \delta_{[2]}) \prod_{j=1}^{n_{[1]}} (1 + \alpha_{[1],j})(1 + \alpha_{[2],[1]})(1 + \alpha_{[2],[2]}),\]

\[\vdots\]

\[C_{[2],n_{[2]}}(S) = t_0(1 + \delta_{[1]})(1 + \delta_{[2]}) \prod_{j=1}^{n_{[1]}} (1 + \alpha_{[1],j}) \prod_{j=1}^{n_{[2]}} (1 + \alpha_{[2],j}).\] (2)

The completion times for jobs in the last group $G_{[m]}$ under $S$ are

$$\begin{align*}
C_{[m],[1]}(S) &= (1 + \delta_{[m]})C_{[m-1],n_{[m-1]}}(S) + \alpha_{[m],[1]}(1 + \delta_{[m]})C_{[m-1],n_{[m-1]}}(S), \\
&= t_0 \prod_{i=1}^{m} (1 + \delta_{[i]}) \prod_{i=1}^{m-1} \prod_{j=1}^{n_{[i]}} (1 + \alpha_{[i],j})(1 + \alpha_{[m],[1]}),
\end{align*}$$

\[C_{[m],[2]}(S) = t_0 \prod_{i=1}^{m} (1 + \delta_{[i]}) \prod_{i=1}^{m-1} \prod_{j=1}^{n_{[i]}} (1 + \alpha_{[i],j})(1 + \alpha_{[m],[1]})(1 + \alpha_{[m],[2]}),\]

\[\vdots\]

\[C_{[m],n_{[m]}}(S) = t_0 \prod_{i=1}^{m} (1 + \delta_{[i]}) \prod_{i=1}^{m-1} \prod_{j=1}^{n_{[i]}} (1 + \alpha_{[i],j})(1 + \alpha_{[m],[1]})(1 + \alpha_{[m],[2]}) \ldots (1 + \alpha_{[m],n_{[m]}}),\]

\[= t_0 \prod_{i=1}^{m} (1 + \delta_{[i]}) \prod_{i=1}^{m} \prod_{j=1}^{n_{[i]}} (1 + \alpha_{[i],j}).\] (3)

Therefore, the makespan of all jobs is

$$\begin{align*}
C_{\text{max}} &= t_0 \prod_{i=1}^{m} (1 + \delta_{[i]}) \prod_{i=1}^{m} \prod_{j=1}^{n_{[i]}} (1 + \alpha_{[i],j}) \\
&= t_0 \prod_{i=1}^{m} (1 + \delta_{[i]}) \prod_{i=1}^{m} \prod_{j=1}^{n_{[i]}} (1 + \alpha_{[i],j}).
\end{align*}$$ \quad (4)

The last step in Eq. (4) follows since the job sequence in each group does not alter the value of $\prod_{i=1}^{m} \prod_{j=1}^{n_{[i]}} (1 + \alpha_{[i],j})$ and the group sequence does not alter the value of $\prod_{i=1}^{m} (1 + \delta_{[i]})$. Thus, the makespan is influenced by neither the group order nor the job order in each group. \quad \square

4. Total completion time minimization

In this section, another single-machine group scheduling problem with deterioration consideration is studied. Once again, the actual job processing times and the actual group setup times are simple linear functions of their starting times. The objective is to find a schedule that minimizes the total completion time of all jobs. The following theorem provides the optimal job and group sequences.
Theorem 2. For the $1/G$, $s_i = \delta_i t$, $p_{ij} = x_{ij} t / \sum C_{ij}$ problem, the optimal schedule is obtained if jobs in each group are ordered according to the smallest deteriorating rate first (SDR) rule:

$$x_{i(1)} \leq x_{i(2)} \leq \cdots \leq x_{i(n_i)}, \quad i = 1, 2, \ldots, m,$$

and groups are arranged in non-decreasing order of

$$(1 + \delta_i) \prod_{j=1}^{n_i} (1 + x_{ij}) - 1 \quad \frac{(1 + \delta_i) \prod_{j=1}^{k} (1 + x_{i(j)})}{(1 + \delta_i) \prod_{j=1}^{k} (1 + x_{i(j)})}.$$

Proof. We prove both conditions by contradiction. Consider an optimal schedule $S$ where the jobs in the same group do not follow the SDR rule. In this schedule, there must be at least one group $G_i$ with at least two adjacent jobs, say $J_{iu}$ followed by $J_{iv}$, such that $x_{iu} > x_{iv}$. Furthermore, we assume that $t$ is the starting time for $J_{iu}$ in $S$. We now perform an adjacent pairwise interchange of $J_{iu}$ and $J_{iv}$, leaving all other jobs in their original positions, to derive a new sequence $S'$. Under $S$, we have

$$C_{iu}(S) = t(1 + x_{iu}),$$
$$C_{iv}(S) = t(1 + x_{iu})(1 + x_{iv}),$$

whereas under $S'$, we obtain

$$C_{iv}(S') = t(1 + x_{iv}),$$
$$C_{iu}(S') = t(1 + x_{iu})(1 + x_{iv}).$$

Furthermore, the completion times of the jobs processed before $J_{iu}$ and $J_{iv}$ are the same since they are processed in the same order. In addition, the completion times of the jobs processed after $J_{iu}$ and $J_{iv}$ are the same since $C_{iv}(S) = C_{iu}(S')$. Thus, we have

$$\sum C_{ij}(S) - \sum C_{ij}(S') = C_{iu}(S) + C_{iv}(S) - C_{iv}(S') - C_{iu}(S')$$

$$= t(x_{iu} - x_{iv})$$

$$> 0,$$

since $x_{iu} > x_{iv}$. It follows that the total completion time of jobs under $S$ is strictly greater than that under $S'$. This contradicts the optimality of $S$ and proves that the jobs in each group are ordered according to SDR rule.

Next, consider an optimal schedule $Q$ in which the group order does not follow the specified rule. In this schedule there must be at least two adjacent groups, $G_i$ followed by $G_j$, such that

$$\frac{(1 + \delta_i) \prod_{l=1}^{n_i} (1 + x_{il}) - 1}{(1 + \delta_i) \prod_{l=1}^{k} (1 + x_{i(l)})} > \frac{(1 + \delta_j) \prod_{l=1}^{n_j} (1 + x_{jl}) - 1}{(1 + \delta_j) \prod_{l=1}^{k} (1 + x_{j(l)})}.$$  

(5)

Furthermore, we assume that $t$ is the starting time to process group $G_i$ in $Q$. We now perform an adjacent pairwise interchange of groups $G_i$ and $G_j$, leaving all other groups in their original order, to derive a new group sequence $Q'$. Under $Q$, the completion time for the $k$th job in group $G_i$ is

$$C_{i[k]}(Q) = t(1 + \delta_i) \prod_{l=1}^{k} (1 + x_{i(l)}),$$

and the completion time for the $k$th job in group $G_j$ is

$$C_{j[k]}(Q) = t(1 + \delta_j)(1 + \delta_j) \prod_{l=1}^{n_i} (1 + x_{i(l)}) \prod_{l=1}^{k} (1 + x_{j(l)}).$$  

(6)
In contrast, under $Q'$ the completion time for the $k$th job in group $G_j$ is

$$C_{j[k]}(Q') = t(1 + \delta_j) \prod_{l=1}^{k} (1 + \alpha_{j(l)}),$$

and the completion time for the $k$th job in group $G_i$ is

$$C_{i[k]}(Q') = t(1 + \delta_i)(1 + \delta_j) \prod_{l=1}^{n_j} (1 + \alpha_{j(l)}) \prod_{l=1}^{k} (1 + \alpha_{i(l)}).$$

(7)

From Eqs. (6) and (7), the completion time for the last job of $G_j$ in $Q$ is equal to that of $G_i$ in $Q'$. This also implies that the completion times for the remaining jobs after group $G_j$ in $Q$ and those after group $G_i$ in $Q'$ are the same. Therefore, the difference in total completion times between $Q$ and $Q'$ is

$$\sum C_{ij}(Q) - \sum C_{ij}(Q') = \sum_{k} C_{i[k]}(Q) + \sum_{k} C_{j[k]}(Q) - \sum_{k} C_{i[k]}(Q') - \sum_{k} C_{j[k]}(Q')$$

$$= t(1 + \delta_j) \left( \sum_{k=1}^{n_j} \prod_{l=1}^{k} (1 + \alpha_{j(l)}) \right) \left( \prod_{l=1}^{n_j} (1 + \alpha_{i(l)}) - 1 \right)$$

$$- t(1 + \delta_i) \left( \sum_{k=1}^{n_i} \prod_{l=1}^{k} (1 + \alpha_{i(l)}) \right) \left( \prod_{l=1}^{n_j} (1 + \alpha_{j(l)}) - 1 \right)$$

$$> 0,$$

based on Eq. (5). It follows that the total completion time of jobs in groups $G_i$ and $G_j$ under $Q$ is strictly greater than that under $Q'$. This contradicts the optimality of $Q$ and proves that groups are arranged in non-decreasing order of

$$\frac{(1 + \delta_i) \prod_{j=1}^{n_j} (1 + \alpha_{ij}) - 1}{(1 + \delta_i) \sum_{k=1}^{n_i} \prod_{j=1}^{k} (1 + \alpha_{i(j)})}.$$

This proves the theorem. $\square$

Using Theorem 2, a simple algorithm to determine an optimal schedule for the $1/G$, $s_i = \delta_it$, $p_{ij} = \alpha_{ij}t/\sum C_{ij}$ problem is described as follows.

**Algorithm 1.**

*Step 1:* Arrange the jobs of each group in non-decreasing order of $\alpha_{ij}$, i.e., $\alpha_{i(1)} \leq \alpha_{i(2)} \leq \cdots \leq \alpha_{i(n_i)}$, $i = 1, 2, \ldots, m$.

*Step 2:* Arrange the groups in non-decreasing order of

$$\frac{(1 + \delta_i) \prod_{j=1}^{n_j} (1 + \alpha_{ij}) - 1}{(1 + \delta_i) \sum_{k=1}^{n_i} \prod_{j=1}^{k} (1 + \alpha_{i(j)})}.$$

It is obvious that the complexity of obtaining the optimal job sequence within group $G_i$ is $O(n_i \log n_i)$ and that of obtaining the optimal group sequence is $O(m \log m)$. Hence, the complexity of Algorithm 1 is at most $O(n \log n)$, where $n_1 + n_2 + \cdots + n_m = n$. We illustrate the algorithm below.

**Example 1.** Consider a single-machine group scheduling problem with seven jobs divided into three groups. The deteriorating rates for each job and the deteriorating rates for each group setup time are shown in Table 1.

**Solution.** According to the algorithm, the optimal schedule is obtained as follows:

*Step 1:* According to the first condition of Theorem 2, the optimal job sequences in groups $G_1$, $G_2$, and $G_3$ are $(J_{13}, J_{12}, J_{11})$, $(J_{22}, J_{21})$, and $(J_{31}, J_{32})$, respectively.
Table 1
An illustrative example

<table>
<thead>
<tr>
<th>Group</th>
<th>$G_1$</th>
<th>$J_{11}$</th>
<th>$J_{12}$</th>
<th>$J_{13}$</th>
<th>$G_2$</th>
<th>$J_{21}$</th>
<th>$J_{22}$</th>
<th>$G_3$</th>
<th>$J_{31}$</th>
<th>$J_{32}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job code</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job deteriorating rate</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
<td>0.4</td>
<td>0.1</td>
<td>0.2</td>
<td>0.4</td>
</tr>
<tr>
<td>Group deteriorating rate</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td>0.3</td>
<td></td>
<td></td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Step 2:** Next, we compute the following values for each group:

\[
\frac{(1 + \delta_1)\prod_{j=1}^{n_1}(1 + \alpha_{1j}) - 1}{(1 + \delta_1)\sum_{k=1}^{n_1}\prod_{j=1}^{k}(1 + \alpha_{1j})} = 0.22,
\]

\[
\frac{(1 + \delta_2)\prod_{j=1}^{n_2}(1 + \alpha_{2j}) - 1}{(1 + \delta_2)\sum_{k=1}^{n_2}\prod_{j=1}^{k}(1 + \alpha_{2j})} = 0.29,
\]

\[
\frac{(1 + \delta_3)\prod_{j=1}^{n_3}(1 + \alpha_{3j}) - 1}{(1 + \delta_3)\sum_{k=1}^{n_3}\prod_{j=1}^{k}(1 + \alpha_{3j})} = 0.27.
\]

Since $0.22 < 0.27 < 0.29$, the optimal group sequence is $(G_1, G_3, G_2)$. Therefore, the optimal schedule is $(J_{13}, J_{12}, J_{11}, J_{31}, J_{32}, J_{22}, J_{21})$, and the total completion time for this optimal sequence is 24.61.

5. Conclusions

Group technology has not been discussed in the context of deteriorating job scheduling problems until recently. Instead, group setup times are assumed to be constant. In reality, process setup or preparation often requires more time as food quality deteriorates or a patient’s condition worsens. We consider a situation where both setup and job processing times are lengthened as jobs wait to be processed. Specifically, we investigate single-machine group scheduling problems where group setup times and job processing times are both increasing functions of their starting times. We show that two objectives, i.e., minimizing the makespan and the total completion time, remain polynomially solvable when deterioration is considered.

In future research, we expect to explore more general deterioration models of group scheduling problems and extend the problems to multiple machine settings.

Acknowledgments

The authors are grateful to the editor and the referees, whose constructive comments have led to a substantial improvement in the presentation of the paper. This work was partly supported by the National Science Council of Taiwan, Republic of Taiwan, under Grant number NSC 94-2213-E-035-020.

References


