Single Machine Group Scheduling Problems with the Effects of Deterioration and Learning

YAN Yang[1] WANG Da-Zhi[1] WANG Ding-Wei[1] W. H. Ip[2] WANG Hong-Feng[1]

Abstract This paper studies the single-machine scheduling problem with the effects of deterioration and learning under group consumption, where the processing time of a job is defined by the function of the starting time and position in the group. Based on the analysis of properties and polynomial algorithms, it can be shown that both the single-machine makespan minimization problem and the total resource minimization problem under the group consumption are polynomially solvable, even though the deterioration and learning effect on job processing time is introduced.

Key words Single-machine, scheduling, deterioration, learning effect, makespan

Scheduling plays an important role in many real-world applications, such as avionics, communications, signal processing, routing, industrial control, operations research, production planning, project management, process scheduling in operating systems, class arrangement, grid computing and so on. Many different techniques have been presented for solving the scheduling problems.

In the classical scheduling problem, it is assumed that the job processing time is fixed and known as the time span from the first job to be processed to the last one. However, this assumption may be unrealistic in many cases, as supported by some industrial empirical studies via demonstrating that the unit cost declines as the firm produces more products and gains knowledge or experience. For instance, Biskup[1] pointed out that the repeated processing of similar tasks continuously improves the worker skills; workers are able to perform setup, deal with machine operations and software, or handle raw materials and components fast. This phenomenon is known as the "learning effect" in the literature. In [2], two special flowshop scheduling models of job processing time characterized by position-dependent function were proved to be polynomially solvable.

Recently, the time dependent learning effect was investigated in [3], where the actual job processing time was designed as the function of the total normal processing time of the previous scheduled jobs. Kuo[4] proved that shortest processing time (SPT) rule was optimal for the total completion time minimization problem. Wang solved the other three problems, whose objectives were to minimize the weighted sum of completion time, the maximum lateness, and the number of tardy jobs, respectively[5]. Moreover, Kuo[6] extended their model into the group scheduling problem with setup time and they presented the optimal algorithm for the extended model. It is obvious that the time-dependent learning model is more practical because it can reflect the fact that the job learning effect is influenced by the normal processing time of the jobs processed before.

Received March 24, 2008; in revised form February 23, 2009
Supported by the State Key Program of National Natural Science of China(70931001), the Science Fund for Creative Research Group of National Natural Science Foundation of China (60821063), National Science and Technology Support Plan of China (2006BAH02A09), the Science Fund for Youth Scholars of Ministry of Education of China (200801451053), and the Research Committee and the Department of Industrial and Systems Engineering of Hong Kong Polytechnic University Research Grants (G-U323)
1. School of Information Science and Engineering, Northeastern University, Shenyang 110004, P. R. China 2. Department of Industrial and Systems Engineering, Hong Kong Polytechnic University, Hong Kong, P. R. China
DOI: 10.3724/SP. J.1004.2009.01290

Empirical studies have shown that the decreasing deteriorating phenomenon is always common in the real life[7−9]. For example, when a worker is assumed to assemble a large number of similar products, the time required by him to assemble one product depends on his knowledge, his skill, and the organization of his working place. After a learning period, the time for him to assemble one product can decrease as a result of the increment of his knowledge, the enhancement of his skill, and the improvement of his working place. Another example is the process of a radar station to recognize aerial threats. When the radar station detects some unidentified objects approaching, the time required to recognize these objects would decrease as they become nearer and nearer. As a result, the later the objects detected, the less time for their recognition.

The phenomena of learning effect and deteriorating jobs occurring simultaneously can be found in many real-life situations. As the manufacturing environment becomes increasingly competitive, the organizations are moving toward the shorter production runs and the more frequent product changes in order to provide greater product variety to the customers. The learning and forgetting that the workers undergo in this environment have thus become increasingly important as they tend to spend more time in rotating among the tasks and responsibilities rather than becoming fully proficient. The workers are often interrupted by the product and process changes, which could cause the decrement of performance. For simplicity, this phenomena is regarded as forgetting in this paper. It should be helpful in improving the accuracy of production planning and productivity estimation to consider the learning and forgetting effects in measuring productivity.

In [10−11], both the deteriorating phenomenon and learning effects were considered in the single machine scheduling problem. Lee[12] considered the single-machine scheduling problem with the deteriorating jobs and the learning effect, whose objective was to minimize makespan and total completion time. He reported the polynomial solutions for both the scheduling problems under the simple linear deterioration. Wang[13] considered the single-machine scheduling problem with the effects of learning and deterioration, where the job processing time was defined by the functions of starting time and position in the sequence. It was shown that even with the introduction of learning effect and deteriorating jobs, the single machine makespan and sum of completion time minimization problem remained polynomially solvable. Wang[14] considered the effects of deterioration and learning in both the single machine and flowshop scheduling models. In

[15], a new model of scheduling problem with learning effect and deteriorating jobs was constructed and discussed, and both the single machine and flowshop cases were analyzed.

However, most studies have ignored the fact that the efficiency of a real production process usually can be increased by grouping various parts and products with similar designs and/or production processes during the production period. This is known as the group technology which has been widely studied in [16−17]. The application of the group technology can bring a lot of benefits. For instance, simplifying the changeovers between different parts can reduce the relevant costs; spending less time on waiting can result in decreasing the work-in-process inventory; the parts tending to move through production in a direct route can reduce the manufacturing lead time; reducing the variability of tasks is helpful in simplifying the worker′s training. Wu[17] discussed two single machine scheduling problems in the context of group technology where the job processing time and setup time were designed as the simple linear functions of starting time. The objectives were to minimize the makespan and the total completion time.

The manufacturing process can always be influenced by the resources. Zhao[18] provided the optimal solutions for the three different scheduling problems to minimize the sum of earliness penalties subject to no tardy jobs, the resource consumption with the makespan constraints, and the makespan with the total resource consumption constraints, respectively. Cheng[19] discussed a single machine scheduling problem to minimize the total resource consumption, with the assumption that the release time of a job was a positive strictly decreasing continuous function of the amount of resources consumed. In [20], the resource constrained scheduling problem was combined into a start time dependent processing time model, and the optimal resource allocation method was given. In many industrial cases, such as the catalyst in the chemical industry or the special lubrication oil during the mechanism processing, the setup time of a group is usually influenced by some kind of resources, which have seldom been considered by the researchers.

In [21], the polynomial time algorithms were presented to find an optimal job sequence and the resource values in order to minimize the total weighted resource consumption subject to the job deadlines. The bi-criterion problem indicated that the time and resource need to be considered together for the resource constrained scheduling problem. Thus, the scheduling model which considers the bi-criterion objectives is more applicable and practical. Cheng[22] gave this kind of bi-criterion model with assumption that the resources for each group are as the same as those consumed by each job.

In this paper, we consider the scheduling model with the effects of deterioration and learning, where the jobs are under group consumption and the setup time of each group is dependent on the resource it consumes. During the practical production process, the same or similar jobs are assigned together to form several groups in order to improve the manufacturing process. Setup time is required if the machine switches from one group to another. And the setup of a group is usually constrained by a kind of resource or several resources.

The remainder of this paper is organized as follows. In Section 1, the problem formulation is given. In Section 2, the makespan minimization problem with the resource constraints is discussed, whereas the total resource consumption minimization problem with the makespan constrains is discussed in Section 3. In Section 4, a computational instance is presented to illustrate the effectiveness of the algorithm. The final section concludes this paper with discussions on the future work.

1 Problem formulation

It is assumed that there are n jobs $J_1, J_2, \cdots, J_n$, which are grouped into m groups and are to be processed on a single machine. All jobs are available at time 0 and processed one by one in the groups on the machine. It is also assumed that there are n_i jobs $J_{i1}, J_{i2}, \cdots, J_{in_i}$ in group G_i. The normal processing time of J_{ij} is p_{ij} and $n_1 + n_2 + \cdots + n_m = n$. Setup time is required if the machine switches from one group to another. The setup time of each group is sequence independent. The setup time s_i of group G_i is dependent on the resource it consumes, i.e.,

$$s_i = f(u_i),\ 0 \le u_i \le \bar{u},\ i = 1, 2, \cdots, m \tag{1}$$

The resource consumption u_i of group G_i is restricted within $[0, \bar{u}]$, where $\bar{u}$ is the upper bound of the resource consumption. f is a decreasing function of the resource that group G_i consumes and its reverse function is f^{-1}.

The processing of a job may not be interrupted. It is assumed that the starting time of J_{ij} is t and its normal processing time is p_{ij}. Its actual processing time is dependent on not only the time it starts, but also its scheduled position in the group. If it is scheduled in the r-th position in group G_i, the actual processing time of J_{ij} is

$$p_{ij}^{[r]} = p_{ij}(b - ct)r^{a_i},\ i = 1, 2, \cdots, m,\ j = 1, 2, \cdots, n_i \tag{2}$$

where a_i is a learning index of group G_i, b and c are two positive constants.

Two versions of objectives are considered. The first one is to minimize the makespan under the resource constrains, whereas the other is to minimize the total resource consumption with the makespan constrains. All the problems considered in this paper will be denoted via using the three-field notation schemas $\alpha|\beta|\gamma$ introduced by Graham[23]. Then, the problems described above can be denoted as follows:

$$1|s_i = f(u_i),\, G,\, p_{ij}(b - ct)r^{a_i},\, \sum u_i \le U|C_{\max} \tag{3}$$

$$1|s_i = f(u_i),\, G,\, p_{ij}(b - ct)r^{a_i},\, C_{\max} \le C|\sum u_i \tag{4}$$

2 Makespan minimization

Lemma 1[14]**.** The makespan of $1| = p_{ij}(b-ct)r^a|C_{\max}$ can be solved by scheduling the jobs in the nonincreasing order of their normal processing time. If the starting time of the first job is t_0 , then the makespan is

$$C_{\max}(t_0|J_1, J_2, \cdots, J_n) = (t_0 - \frac{b}{c})\prod_{j=1}^{n}(1 - cp_j j^a) + \frac{b}{c} \tag{5}$$

Theorem 1. For the problem $1|G, p_{ij}(b - ct)r^{a_i}|C_{\max}$, if the starting time of the first group is $t_0 = 0$, then the makespan is

$$C_{\max} = \sum_{k=1}^{m}\left(s_k \prod_{i=k}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})\right) - \frac{b}{c}\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i}) + \frac{b}{c} \tag{6}$$

Proof. Assume $\pi = [G_{\pi(1)}, G_{\pi(2)}, \cdots, G_{\pi(m)}] = [G_1, G_2, \cdots, G_m]$ is a schedule of the groups. There are n_i jobs in group G_i $(i = 1, \cdots, m)$, and job J_{ij} is processed at j-th position in group G_i. The completion time of group G_1 and group G_2 are

$$C_1 = \left(s_1 - \frac{b}{c}\right)\prod_{j=1}^{n_1}(1-cp_{1j}j^{a_1}) + \frac{b}{c}$$

$$C_2 = \left(C_1 + s_2 - \frac{b}{c}\right)\prod_{j=1}^{n_2}(1-cp_{2j}j^{a_2}) + \frac{b}{c} = \sum_{k=1}^{2}\left(s_k \prod_{i=k}^{2}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})\right) - \frac{b}{c}\prod_{i=1}^{2}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i}) + \frac{b}{c}$$

Assume (6) is true for the l-th group. Then,

$$C_l = \sum_{k=1}^{l}\left(s_k \prod_{i=k}^{l}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})\right) - \frac{b}{c}\prod_{i=1}^{l}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i}) + \frac{b}{c}$$

Considering group G_{l+1}, we have

$$C_{l+1} = \left(C_l + s_{l+1} - \frac{b}{c}\right)\prod_{j=1}^{n_{l+1}}(1-cp_{l+1,j}j^{a_{l+1}}) + \frac{b}{c} = \Big[\sum_{k=1}^{l}\left(s_k \prod_{i=k}^{l}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})\right) - \frac{b}{c}\prod_{i=1}^{l}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i}) + \frac{b}{c} + s_{l+1} - \frac{b}{c}\Big]\times \prod_{j=1}^{n_{l+1}}(1-cp_{l+1,j}j^{a_{l+1}}) - \frac{b}{c} = \sum_{k=1}^{l+1}\left(s_k \prod_{i=k}^{l+1}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})\right) - \frac{b}{c}\prod_{i=1}^{l+1}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i}) + \frac{b}{c}$$

That is, (6) is true for group G_{l+1}.

Based on the induction of assumption, (6) is true for the m-th group. Then,

$$C_{\max} = \sum_{k=1}^{m}\left(s_k \prod_{i=k}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})\right) - \frac{b}{c}\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i}) + \frac{b}{c}$$

□

Property 1. The jobs in each group are scheduled in the nonincreasing order of their normal processing time.

Proof. For any sequence of the groups $\pi = [G_1, G_2, \cdots, G_m]$, we could prove that $C_{\max}$ could be minimized by scheduling the jobs in each group in the nonincreasing order of their normal processing time by the method of adjacent pairwise interchange. □

Property 2. In order to minimize the objective $C_{\max}$, the groups scheduled in the later position are given the priority of resource allocation for any schedule of the groups.

Proof. Without loss of generality, we assume $\pi = [G_1, G_2, \cdots, G_m]$ is a random schedule of all the groups. Based on (6), we first allocate the resource to the setup time s_i with the bigger $\prod_{i=l}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})$. The makespan can be minimized by the same amount of resource. Since $c > 0$, $p_{ij} > 0$, $i = 1, 2, \cdots, m$, $j = 1, 2, \cdots, n_i$, we have $\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i}) < 1$ for any i. Thus,

$$\prod_{i=l}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i}) < \prod_{i=2}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i}) < \cdots < \prod_{i=m}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})$$

Then, the groups scheduled in the later position should be given the priority of resource allocation to minimize the objective $C_{\max}$. □

Property 3. Based on Properties 1 and 2, we can schedule the jobs in each group in the nonincreasing order of their normal processing time and allocate the resource to the setup time of the groups in each position. If the resource amount of the groups in each position is fixed and the jobs are scheduled in the nonincreasing order of their normal processing time in each group, the optimal schedule can be obtained by scheduling the groups in the nondecreasing order of $\rho_{G_i} = \prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})$.

Proof. Without loss of generality, we assume that the resource amount of the group scheduled at the i-th position is denoted as $u^*_{[i]}$ $(i = 1, 2, \cdots, m)$, and that G_l and G_{l+1} are two adjacent groups in the optimal schedule $\pi = [G_1, G_2, \cdots, G_m]$. G_l is scheduled in front of G_{l+1} and $\prod_{j=1}^{n_l}(1-cp_{lj}j^{a_l}) > \prod_{j=1}^{n_{l+1}}(1-cp_{l+1,j}j^{a_{l+1}})$. If changing the sequences of G_l and G_{l+1}, we can obtain a new schedule π'. In π', G_l is scheduled in the $(l+1)$-th position and its resource allocation amount is $u^*_{[l+1]}$; G_{l+1} is scheduled at the l-th position and its resource allocation amount is $u^*_{[l]}$. Then, the objective values of π and π' are

$$C_{\max} = C_m = f(u^*_{[1]})\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i}) + \cdots + f(u^*_{[l-1]})\prod_{i=l-1}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i}) + f(u^*_{[l]})\prod_{i=l}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i}) + f(u^*_{[l+1]})\prod_{i=l+1}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i}) + f(u^*_{[l+2]})\times \prod_{i=l+2}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i}) + \cdots + f(u^*_{[m]})\prod_{j=1}^{n_m}(1-cp_{mj}j^{a_i}) -$$

$$\frac{b}{c}\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})+\frac{b}{c}=$$

$$\sum_{k=1}^{m}f(u_{[k]}^*)\prod_{i=k}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})+K$$

$$C'_{\max}=C'_m=$$

$$f(u_{[1]}^*)\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})+\cdots+f(u_{[l-1]}^*)\times$$

$$\prod_{i=l-1}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})+f(u_{[l]}^*)\prod_{j=1}^{n_{l+1}}(1-cp_{l+1,j}j^{a_{l+1}})\times$$

$$\prod_{j=1}^{n_l}(1-cp_{lj}j^{a_l})\prod_{i=l+2}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})+f(u_{[l+1]}^*)\times$$

$$\prod_{j=1}^{n_l}(1-cp_{lj}j^{a_l})\prod_{i=l+2}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})+f(u_{[l+2]}^*)\times$$

$$\prod_{i=l+2}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})+\cdots+f(u_{[m]}^*)\prod_{j=1}^{n_m}(1-cp_{mj}j^{a_i})-$$

$$\frac{b}{c}\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})+\frac{b}{c}=$$

$$\sum_{k=1}^{l-1}f(u_{[k]}^*)\prod_{i=k}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})+$$

$$f(u_{[l]}^*)\prod_{i=l}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})+$$

$$f(u_{[l+1]}^*)\prod_{j=1}^{n_l}(1-cp_{lj}j^{a_l})\prod_{i=l+2}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})+$$

$$\sum_{k=l+2}^{m}f(u_{[k]}^*)\prod_{i=k}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})+K$$

where $K=-\frac{b}{c}\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})+\frac{b}{c}$. Then, we can obtain

$$C_{\max}-C'_{\max}=f(u_{[l+1]}^*)\prod_{i=l+2}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})\times$$
$$\left(\prod_{j=1}^{n_{l+1}}(1-cp_{l+1,j}j^{a_{l+1}})-\prod_{j=1}^{n_l}(1-cp_{lj}j^{a_l})\right)>0$$

Thus π' excels π, which is contradictory to the optimality of π. If the resource amount of the groups in each position is fixed, the optimal schedule can be obtained by scheduling the groups in the nondecreasing order of $\rho_{G_i}=\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})$. □

Based on Properties 1 ∼ 3, the optimal schedule can be obtained by scheduling the jobs in each group in the nonincreasing order of their normal processing time, scheduling the groups in the nondecreasing order of $\rho_{G_i}=\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})$ and allocating the resource to the groups scheduled in later position.

Property 4. Since the premise is to minimize makespan with the resource constraints, the makespan can be minimized under the condition that $\sum U_i=U$.

According to the above analysis, $1|s_i=f(u_i),G,p_{ij}(b-ct)r^{a_i},\sum u_i\le U|C_{\max}$ can be solved by the following algorithm.

Algorithm 1.

Step 1. Order the jobs in each group in the nonincreasing order of their normal processing time.

Step 2. Order the groups by the nondecreasing order of $\rho_{G_i}=\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})$. For the obtained schedule $\pi=[G_{[1]},G_{[2]},\cdots,G_{[m]}]$, set $u_{[i]}^*=0,i=1,2,\cdots,m,l=m$.

Step 3. Set $u_{[l]}^*=\min\{\overline{u},U\},U=U-u_{[l]}$, and $l=l-1$.

Step 4. If $U=0$ or $l=0$, exit; else go to Step 3.

Obviously, the total time for Algorithm 1 is $\mathrm{O}(n\log n)$.

3 Total resource consumption minimization

For the problem $1|s_i=f(u_i),G,p_{ij}(b-ct)r^{a_i},C_{\max}\le C|\sum u_i$, we can draw the following conclusions based on the above analysis of problem (1). Firstly, the optimal schedule can be obtained by scheduling the jobs in each group in the nonincreasing order of their normal processing time and scheduling the groups in the nondecreasing order of $\rho_{G_i}=\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})$ in order to minimize the resource consumption. Secondly, the total resource consumption can be minimized under the condition that $C_{\max}=C$ since the premise is to minimize the total resource consumption with the makespan constraints. Finally, the resource should be allocated to the groups scheduled in the later position.

If we only allocate the resource to the setup time of group $G_{\pi_{[m]}}$ and the resource allocation amount is 0 for the groups scheduled before it, we have

$$s_{[m]}\prod_{i=m}^{m}\prod_{j=1}^{n_{[i]}}(1-cp_{[i]j}j^{a_{[i]}})+s\sum_{k=1}^{m-1}\prod_{i=k}^{m}\prod_{j=1}^{n_{[i]}}(1-cp_{[i]j}j^{a_{[i]}})=C-K$$

according to (6). Thus, the setup time of $G_{[m]}$ is

$$s_{[m]}=\frac{C-K-s\sum\limits_{k=1}^{m-1}\prod\limits_{i=k}^{m}\prod\limits_{j=1}^{n_i}(1-cp_{[i]j}j^{a_{[i]}})}{\prod\limits_{i=m}^{m}\prod\limits_{j=1}^{n_i}(1-cp_{[i]j}j^{a_{[i]}})}$$

where $s=f(0)$ and $K=-\frac{b}{c}\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1-cp_{ij}j^{a_i})+\frac{b}{c}$. But as we known, the resource consumption $u_{[m]}^*$ of $G_{[m]}$ is restricted within $[0,\bar{u}]$. Thus, we have:

If $s_{[m]}\ge f(0)$, then the restriction that $C_{\max}\le C$ can be satisfied even if the resource allocation amount $u_{[m]}^*$ of $G_{[m]}$ is 0. We do not have to allocate the resource to the groups scheduled before it, and we stop.

If $f(\bar{u})\le s_{[m]}<f(0)$, then first calculate $f^{-1}(s_{[m]})$, the optimal resource allocation amount of $G_{[m]}$ is $u_{[m]}^*=f^{-1}(s_{[m]})$. We do not have to allocate the resource to the former groups, and we stop.

If $s_{[m]}<f(\bar{u})$, then the resource allocation amount of $G_{[m]}$ should be $\bar{u}$, that is, $u_{[m]}^*=\bar{u}$. We have to go on allocating the resource to the groups scheduled before $G_{[m]}$. The resource consumption of $G_{[m]}$ is equal to its upper bound $\bar{u}$. Then, the optimal resource consumption $u_{[m-1]}^*,u_{[m-2]}^*,\cdots,u_{[1]}^*$ of other groups can be calculated by the same method.

According to the above analysis, $1|s_i=f(u_i)$, G, $p_{ij}(b-ct)r^{a_i},C_{\max}\le C|\sum u_i$ can be solved by the following algorithm.

Algorithm 2.

Step 1. Order the jobs in each group in the nonincreasing order of their normal processing time.

Step 2. Order the groups by the nondecreasing order of $\rho_{G_i} = \prod_{j=1}^{n_i}(1 - cp_{ij}j^{a_i})$. Let $\pi = [G_{[1]}, G_{[2]}, \cdots, G_{[m]}]$ be the obtained schedule.

Step 3. Set $u^*_{[i]} = 0, i = 1, 2, \cdots, m, U = 0, s = f(0), s' = f(\overline{u}), l = m$; calculate $K = -\frac{b}{c}\prod_{i=1}^{m}\prod_{j=1}^{n_i}(1 - cp_{ij}j^{a_i}) + \frac{b}{c}$; and set $C' = C - K$.

Step 4. Calculate

$$s_{[l]} = \frac{C' - s\sum_{k=1}^{l-1}\prod_{i=k}^{m}\prod_{j=1}^{n_i}(1 - cp_{[i]j}j^{a_{[i]}})}{\prod_{i=l}^{m}\prod_{j=1}^{n_i}(1 - cp_{[i]j}j^{a_{[i]}})}$$

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

Step 5. If $s_{[l]} \geq f(0)$, then exit; if $f(\overline{u}) \leq s_{[l]} < f(0)$, then set $u^*_{[l]} = f^{-1}(s_{[l]}), U = U + u^*_{[l]}$, and exit, and the optimal resource allocation is $u^* = [u^*_1, u^*_2, \cdots, u^*_m]$; if $s_{[l]} < f(\overline{u})$, then go to Step 6.

Step 6. Set $u^*_{[l]} = \bar{u}, U = U + u^*_{[l]}, C' = C' - s'\prod_{i=l}^{m}\prod_{j=1}^{n_i}(1 - cp_{ij}j^{a_i})$, and $l = l - 1$. If $l < 0$, then exit, there is no feasible solution; else go to Step 4.

The total time for Steps 1 and 2 is $\mathrm{O}(n\log n)$ and the total time for Steps 3 $\sim$ 6 is $\mathrm{O}(ng(n))$ if the total time for f^{-1} is $g(n)$. Obviously, the total time for Algorithm 2 is $\max\{\mathrm{O}(n\log n, \mathrm{O}(ng(n)))\}$.

4 Computational example

Example. Consider a single-machine group-scheduling problem with seven jobs divided into three groups. The setup time s_i of group G_i is dependent on the resource it consumes: $s_i = 20 - 0.12u_i^2 - 0.08u_i$, $0 \leq u_i \leq \bar{u}$, $i = 1, 2, 3$. If J_{ij} is scheduled in the r-th position in group G_i, the actual processing time of J_{ij} is $p^{[r]}_{ij} = p_{ij}(b - ct)r^{a_i}$, $i = 1, 2, 3$, $j = 1, 2, \cdots, n_i$. It is assumed that $b = 1, c = 0.004$, $U = 9$, and $\bar{u} = 5$. The basic processing time for each job and the learning index of each group are illustrated in Table 1. The optimization objective is to minimize the makespan with the resource constraints.

Table 1 An illustrative example

G̃roup	G_1			G_2		G_3	
Job code	J_{11}	J_{12}	J_{13}	J_{21}	J_{22}	J_{31}	J_{32}
Normal processing time p_{ij} of J_{ij}	42	16	37	21	49	37	43
Group learning index		−0.05		−0.10		−0.08	

Solution. The procedure to obtain the optimal schedule is as follows:

Step 1. Sorting the jobs within the same group according to the nonincreasing order of their normal processing time, we have the optimal job sequences in G_1, G_2, and G_3 are $[J_{11}, J_{13}, J_{12}]$, $[J_{22}, J_{21}]$, and $[J_{32}, J_{31}]$, respectively.

Step 2. Calculate ρ_{G_i} for each group: $\rho_{G_1} = \prod_{j=1}^{n_1}(1 - cp_{1j}j^{a_1}) = (1 - 0.004\times42\times1^{-0.05})(1 - 0.004\times37\times2^{-0.05})(1 - 0.004\times16\times3^{-0.05}) \approx 0.662$.

In a similar way, we can get $\rho_{G_2} = 0.741$ and $\rho_{G_3} = 0.712$. Sort the group in the non-decreasing order of the ρ_{G_i}. Thus, the optimal group sequence is $[G_1, G_3, G_2]$. Set $u^*_{[i]} = 0$, $i = 1, 2, 3$ and $l = 3$.

Step 3. Set $u^*_{[3]} = \min\{\overline{u}, U\} = \min\{5, 9\} = 5, U = U - u_{[3]} = 4, l = 3 - 1 = 2$; $u^*_{[2]} = \min\{\overline{u}, U\} = \min\{5, 4\} = 4, U = U - u_{[2]} = 0$; exit. That is, $u^*_1 = 0, u^*_2 = 5$ and $u^*_3 = 4$.

Therefore, the optimal resource allocation is u^*=[0, 5, 4] and the optimal schedule is $[J_{11}$, J_{13}, J_{12}, J_{32}, J_{31}, J_{22}, $J_{21}]$ Then, according to (6), the makespan for this optimal resource allocation and sequence is 191.83.

5 Conclusion

In this paper, a new single-machine scheduling model, which considers the effects of deterioration and learning under the group consumption, was investigated. In the investigated model, each processing job is dependent on both its starting time and its scheduled position in the corresponding group, and the setup time of each group is dependent on the resource it consumes. The makespan minimization problem with total resource consumption constraints and the total resource consumption minimization problem with the makespan constraints were discussed, respectively. For both discussed problems, the properties and polynomial algorithms were also presented, respectively. This paper finally report the results of applying the proposed algorithms to an illustrative instance.

Future work will focus on investigating other objectives in the single-machine scheduling problem, and then other group problems in the multi-machine and job-shop scheduling fields.

References

1 Biskup D. Single-machine scheduling with learning considerations. *European Journal of Operational Research*, 1999, **115**(1): 173−178

2 Wang J B, Xia Z Q. Flow-shop scheduling with a learning effect. *The Journal of the Operational Research Society*, 2005, **56**(11): 1325−1330

3 Kuo W H, Yang D L. Minimizing the makespan in a single-machine scheduling problem with a time-based learning effect. *Information Processing Letters*, 2006, **97**(2): 64−67

4 Kuo W H, Yang D L. Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect. *European Journal of Operational Research*, 2006, **174**(2): 1184−1190

5 Wang J B, Ng C T, Cheng T C E, Liu L L. Single-machine scheduling with a time-dependent learning effect. *International Journal of Production Economics*, 2008, **111**(2): 802−811

6 Kuo W H, Yang D L. Single-machine group scheduling with a time-dependent learning effect. *Computers and Operations Research*, 2006, **33**(8): 2099−2112

7 Ng C T, Cheng T C E, Bachman A, Janiak A. Three scheduling problems with deteriorating jobs to minimize the total completion time. *Information Processing Letters*, 2002, **81**(6): 327−333

8 Wang J B, Wang M Z, Xia Z Q. Single-machine scheduling to minimize makespan under linear deterioration. *International Journal of Pure and Applied Mathematics*, 2004, **13**(4): 529−540

9 Wang J B, Xia Z Q. Flow shop scheduling with deteriorating jobs under dominating machines. *Omega*, 2006, **34**(4): 327−336

10 Wang X L, Cheng T C E. Single-machine scheduling with deteriorating jobs and learning effects to minimize the makespan. *European Journal of Operational Research*, 2007, **178**(1): 57−70

11 Van Wanssenhove L N, Potts C N. Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity. *Journal of the Operational Research Society*, 1992, **43**(5): 395−406

12 Lee W C. A note on deteriorating jobs and learning in single-machine scheduling problems. *International Journal of Business and Economics*, 2004, **3**(1): 83−89

13 Wang J B. Single-machine scheduling problems with the effects of learning and deterioration. *Omega*, 2007, **35**(4): 397−402

14 Wang J B, Cheng T C E. Scheduling problems with the effects of deterioration and learning. *Asia-Pacific Journal of Operational Research*, 2007, **24**(2): 245−261

15 Wang J B. A note on scheduling problems with learning effect and deteriorating jobs. *International Journal of Systems Science*, 2006, **37**(12): 827−833

16 Ng C T, Cheng T C E, Janiak A, Kovalyov M Y. Group scheduling with controllable setup and processing times: minimizing total weighted completion time. *Annals of Operations Research*, 2005, **133**(1-4): 163−174

17 Wu C C, Shiau Y R, Lee W C. Single-machine group scheduling problems with deterioration consideration. *Computers and Operations Research*, 2008, **35**(5): 1652−1659

18 Zhao C L, Tang H Y. Single machine scheduling problems with deteriorating jobs. *Applied Mathematics and Computation*, 2005, **161**(33): 865−874

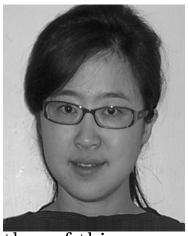
19 Cheng T C E, Janiak A. Resource optimal control in some single-machine scheduling problems. *IEEE Transactions on Automatic Control*, 1994, **39**(6): 1243−1246

20 Zhao Chuan-Li, Zhang Qing-Ling, Tang Heng-Yong. Single machine scheduling with linear processing times. *Acta Automatica Sinica*, 2003, **29**(5): 703−708

21 Janiak A, Kovalyov M Y, Marie-Claude P. Single machine group scheduling with resource dependent setup and processing times. *European Journal of Operational Research*, 2005, **162**(1): 112−121

22 Cheng T C E, Janiak A, Kovalyov M Y. Single machine batch scheduling with resource dependent setup and processing times. *European Journal of Operational Research*, 2001, **135**(1): 177−183

23 Graham R L, Lawler E L, Lenstra J K, Rinnooy Kan A H G. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 1979, **5**: 287−326



YAN Yang Ph. D. candidate at the School of Information Science and Engineering, Northeastern University. She received her B. Sc. degree in mathematics and applied mathematics in 2004, and the M. Sc. degree in operations research in 2007 from Shenyang Normal University. Her research interest covers dynamic evolutionary computation, dynamic scheduling, and scheduling theory. Corresponding author of this paper. E-mail: yanyangmail@163.com

WANG Da-Zhi Received his B. Sc. degree in automation and the Ph. D. degree in systems engineering from Northeastern University in 2001 and 2009, respectively. He is currently a lecturer at the School of Information Science and Engineering, Northeastern University. His research interest covers complex industrial process modeling, control and optimizing, evolutionary computation and application.
E-mail: dzwang@gmail.com

WANG Ding-Wei Received his M.Sc. degree in systems engineering from Huazhong University of Science and Technology in 1984, and the Ph. D. degree in control theory and application from Northeastern University in 1993. He is currently a professor at the Institute of Systems Engineering, Northeastern University. His research interest covers modeling of optimization of complex systems, evolutionary computation, and soft computing.
E-mail: dwwang@mail.neu.edu.cn

W. H. Ip Received his M. Sc. degree from Cranfield University, UK in 1983, the M.B.A. degree from Brunel University, UK in 1989, and the Ph. D. degree in manufacturing engineering from Loughborough University, UK in 1993. He is currently an associate professor with the Department of Industrial and Systems Engineering, Hong Kong Polytechnic University. His research interest covers modeling of the application of intelligent algorithms in planning and scheduling, evolutionary computing, and computer visions.
E-mail: mfwhip@inet.polyu.edu.hk

WANG Hong-Feng Received his B. Sc. degree in industrial engineering and M. Sc. and Ph. D. degrees in systems engineering from Northeastern University in 2001, 2004, and 2007, respectively. He is currently a lecturer at the Institute of Systems Engineering, Northeastern University. His research interest covers evolutionary algorithms, memetic computing, computational intelligence in dynamic environments, production scheduling, and supply chain management. E-mail: hfwang@mail.neu.edu.cn