Journal of Southeast University (English Edition) Vol. 28, No. 3, pp. 354 – 359 Sept. 2012 ISSN 1003—7985

Group scheduling with general position-dependent effect

Yu Xianyu[1,2] Zhang Yulin[1] Zhang Yuhuan[1]

([1]School of Economics and Management, Southeast University, Nanjing 211189, China)

([2]School of Science, East China Institute of Technology, Nanchang 330000, China)

Abstract: In order to investigate more realistic group scheduling problems with position-dependent effects, the model of general position-dependent group scheduling is proposed, where the actual group setup times and actual processing times are described by general functions of the normal group setup time and position in the sequence. These general functions are not assumed to have specific function structures, and are not restricted to be monotone. By mathematical analysis and proof, each considered problem is decomposed into a group scheduling process and a job scheduling process, and each scheduling process is transferred into the classic assignment problem or the classic single-machine sequence problem, and then the computational complexity to solve the considered problem is analyzed. Analysis results show that, even with general position-dependent job processing times, both the single machine makespan minimization group scheduling problems and the parallel-machine total load minimization group scheduling problems remain polynomially solvable.

Key words: group scheduling; position-dependent; makespan; total load

doi: 10.3969/j. issn. 1003 – 7985. 2012. 03. 018

Recently, scheduling problems with the learning effect or the aging effect have attracted much attention. In a learning environment, the later a given job is scheduled in the sequence, the shorter its processing time; while in an aging environment, the later a given job is scheduled in the sequence, the longer its processing time. Many researchers have devoted themselves to addressing the learning or aging effect in scheduling under different machine environments[1–5].

In actual manufacturing, grouping similar products into families helps increase the efficiency of operations and decrease the requirement of facilities. This concept is known as group technology[6]. In group technology, the jobs are classified into groups according to the similar production requirements; there are no machine setups between two consecutively scheduled jobs from the same group. Many recent studies have been conducted to address the group technology on scheduling problems with learning or aging effect group setup time. Among these, several different linear time-dependent aging models of the group setup time were proposed in Refs. [7 – 9]. The group setup time in Ref. [10] was assumed to follow a simple linear time-dependent learning model. Lee et al. [11–12] considered a position-dependent learning setup time model, which is described by an exponential function of normal group setup time and the position in the group sequence. Wang et al. [13] gave a revised model of Zhang et al. [12], and showed that some single machine scheduling problems are still polynomially solvable under the revised model.

Note that all of the papers mentioned above concerning learning effects on scheduling problems are based on specific learning or aging functions. With the motivation of this point, this paper aims to study the scheduling problem with a general position-dependent group setup time model, where the general group setup time function has no specific function structure, and is not restricted to follow a learning or aging model. To the best of our knowledge, this kind of group setup time model is not found at the existing research about group scheduling. In many actual scenarios, workers gain the experience from producing products, while the fatigue of workers and the abrasion of machines are accumulating at the same time. It is obvious that learning and aging effects are sometimes concurrent and difficult to be described by one or two specific functions. Therefore, the general position-dependent group setup time model considered in this paper is worth being studied in both theory and practice.

1 Problem Formulation

The problems under study can be formally described as follows. There are n jobs to be classified into l groups to be processed on m parallel machines. All the jobs and machines are available at time zero, and are not allowed to be preempted in the processing of producing a job. The jobs in the same group are processed consecutively. If the machine switches to process from one group to another, a group setup time is required. The group setup time is assumed to follow the following general position-dependent

Received 2011-12-20.

Biographies: Yu Xianyu (1980—), male, graduate, lecturer; Zhang Yulin (corresponding author), male, doctor, professor, zhangyl@seu.edu.cn.

Foundation items: The National Natural Science Foundation of China (No. 71171046), the Scientific Research Innovation Project for College Graduates in Jiangsu Province (No. CXLX_0162).

Citation: Yu Xianyu, Zhang Yulin, Zhang Yuhuan. Group scheduling with general position-dependent effect[J]. Journal of Southeast University (English Edition), 2012, 28(3): 354 – 359. [doi: 10. 3969/j. issn. 1003 – 7985. 2012. 03. 018]

model. If group G_i is scheduled in the r-th position in the group sequence on parallel machine k, its actual setup time is defined by

$$s_{ik}^t = s_i(i, t) \qquad i, t = 1, 2, \ldots, l \tag{1}$$

where s_{ik}^t is a general function of the group number i and position number t in the group sequence, and it is not restricted to be monotone. For different groups, the corresponding functions can be different. For simplicity, we use PDG to denote the position-dependent group setup time model.

The job processing time model considered in this paper concerns the position-dependent learning effect whereby if job J_{ij} is scheduled in the r-th position of group G_i, its actual processing time is defined by

$$p_{ij}^r = p_{ij}(i, j, r) \qquad i = 1, 2, \ldots, l;\ j = 1, 2, \ldots, n_i;\ r = 1, 2, \ldots, n_i \tag{2}$$

which is a general function of the group number i, job number j and position number r in group sequence. Similar to the definition of the actual group setup time s_{ik}^t, the actual processing time p_{ij}^r is not restricted to be monotone. Note that Mosheiov[14] proposed a similar job processing time model: $p_j^r = p(j, r)$, where p_j^r is the actual processing time of the job J_j scheduled in the r-th position in the job sequence. Since the function $p_{ij}(i, j, r)$ can vary from different jobs, the actual job processing time model considered in this paper is an extended version of the model introduced in Ref. [14]. For simplicity, we use PDP to denote this kind of general position-dependent processing time model.

Let $C_{\max}^k$ denote the the maximum job completion time on machine M_k, and let L_T denote the total load of all machines, i. e., $L_T = \sum_{k=1}^{m} C_{\max}^k$. For single-machine scheduling, $C_{\max}$ denotes the maximum job completion time on the machine, and it is obvious that $L_T = C_{\max}$. For parallel machines scheduling, the objective is to minimize the total load of all the machines. While for the single-machine scheduling problem, the objective is to minimize the makespan. By using the three-field notation scheme for the scheduling problem introduced by Graham et al. [15], the single machine makespan minimization group scheduling problem can be denoted as $1 \mid \text{PDG, PDP} \mid C_{\max}$, while the parallel-machine total load minimization group scheduling problem can be denoted as $P_m \mid \text{PDG, PDP} \mid L_T$.

For the $1 \mid \text{PDG, PDP} \mid C_{\max}$ problem, we use s_i^t to denote the actual setup time of group G_i scheduled in the t-th position in the group sequence (i. e., $s_i^t = s_i(i, t)$). Let $s^{[i]}$ be the actual setup time scheduled in the i-th position in the group sequence. Let $p_{[i]}^{[r]}$ be the actual processing time of the job scheduled in the r-th position in the i-th group, where there are $n_{[i]}$ jobs in the group. We can obtain the makespan of the problem $1 \mid \text{PDG, PDP} \mid C_{\max}$ as follows:

$$\begin{aligned} C_{\max} &= s^{[1]} + (p_{[1]}^{[1]} + p_{[1]}^{[2]} + \ldots + p_{[1]}^{n_{[1]}}) + s^{[2]} + \\ &\quad (p_{[2]}^{[1]} + p_{[2]}^{[2]} + \ldots + p_{[2]}^{n_{[2]}}) + \ldots + s^{[l]} + \\ &\quad (p_{[l]}^{[1]} + p_{[l]}^{[2]} + \ldots + p_{[l]}^{n_{[l]}}) = \\ &\quad \sum_{j=1}^{l} s^{[j]} + \sum_{i=1}^{l} \sum_{r=1}^{n_{[i]}} p_{[i]}^{[r]} \end{aligned}$$

We can reformulate the makespan $C_{\max}$ as follows:

$$C_{\max} = \sum_{j=1}^{l} \sum_{t=1}^{l} s_i^t x_{it} + \sum_{i=1}^{l} \sum_{j=1}^{n_i} \sum_{r=1}^{n_i} p_{ij}^r x_{jr} \tag{3}$$

where x_{it} is a 0/1 variable such that $x_{it} = 1$ if group G_i is the t-th group to be processed, and $x_{it} = 0$, otherwise; x_{jr} is another 0/1 variable such that $x_{jr} = 1$ if job J_{ij} is scheduled in the r-th position in group G_i, and $x_{jr} = 0$, otherwise.

With the similar method, we can obtain the total load of the $P_m \mid \text{PDG, PDP} \mid L_T$ problem as follows:

$$L_T = \sum_{i=1}^{l} \sum_{k=1}^{m} \sum_{t=1}^{l} s_{ik}^t \mathrm{x}_{ikt} + \sum_{i=1}^{l} \sum_{j=1}^{n_i} \sum_{r=1}^{n_i} p_{ij}^r x_{jr} \tag{4}$$

where $x_{ikt} = 1$ on the condition that group G_i is scheduled in the t-th group position on machine M_k, and $x_{it} = 0$, otherwise.

2 Main Results

In this section, we first investigate the single-machine group scheduling problem. We propose several lemmas, which are useful in the following theorems.

Lemma 1 The single machine scheduling problem of minimizing $\sum_{i=1}^{l} \sum_{t=1}^{l} s_i^t x_{it}$ can be optimally solved in $O(l^3)$ time.

Proof The problem of minimizing $\sum_{i=1}^{l} \sum_{t=1}^{l} s_i^t x_{it}$ can be transferred as the following assignment problem with $l \times l$ group positions:

$$\min \sum_{i=1}^{l} \sum_{t=1}^{l} s_i^t x_{it} \tag{5}$$

s. t.

$$\sum_{i=1}^{l} x_{it} = 1 \qquad i = 1, 2, \ldots, l \tag{6}$$

$$\sum_{t=1}^{l} x_{it} = 1 \qquad t = 1, 2, \ldots, l \tag{7}$$

$$x_{it} = 1 \text{ or } 0 \qquad i, t = 1, 2, \ldots, l \tag{8}$$

where $x_{it} = 1$ if group G_i is the t-th group to be processed, and $x_{it} = 0$, otherwise. It is well-known that an assignment problem with $l \times l$ group-positions can be optimally solved in $O(l^3)$ time (by the Hungarian method, see, e. g. Ref. [16]). Thus, the computational complexity of solving the single machine scheduling problem of minimizing $\sum_{i=1}^{l}\sum_{t=1}^{l} = s_i^t x_{it}$ is $O(l^3)$.

Lemma 2 In the group G_i, the scheduling problem of minimizing $\sum_{j=1}^{n_i}\sum_{r=1}^{n_i} p_{ij}^r x_{jr}$ can be optimally solved in $O(n_i^3)$ time.

Proof Similar to the proof of Lemma 1, the problem of minimizing $\sum_{j=1}^{n_i}\sum_{r=1}^{n_i} p_{ij}^r x_{jr}$ can be transferred as the following assignment problem with $n_i \times n_i$ group positions:

$$\min \sum_{j=1}^{n_i}\sum_{r=1}^{n_i} p_{ij}^r x_{jr} \tag{9}$$

s. t.

$$\sum_{j=1}^{n_i} x_{jr} = 1 \qquad j = 1, 2, \ldots, n_i \tag{10}$$

$$\sum_{r=1}^{n_i} x_{jr} = 1 \qquad r = 1, 2, \ldots, n_i \tag{11}$$

$$x_{jr} = 1 \text{ or } 0 \qquad j, r = 1, 2, \ldots, n_i \tag{12}$$

where $x_{jr} = 1$ if job J_{ij} is scheduled in the r-th position in group G_i, and $x_{jr} = 0$, otherwise. For the $n_i \times n_i$ assignment problem (9)-(12), the problem of minimizing $\sum_{j=1}^{n_i}\sum_{r=1}^{n_i} p_{ij}^r x_{jr}$ can be optimally solved in $O(n_i^3)$ time (by the Hungarian Method, see, e. g. Ref. [16]).

Theorem 1 The $1 \mid \mathrm{PDG}, \mathrm{PDP} \mid C_{\max}$ problem can be optimally solved in $O(n^3)$ time.

Proof Based on Eq. (3) and Lemmas 1 and 2, the $1 \mid \mathrm{PDG}, \mathrm{PDP} \mid C_{\max}$ problem can be optimally solved in two steps. The first step schedules all the groups according to the solution of the assignment problem (5)-(8). The second step schedules all the jobs in group $G_i (i = 1, 2, \ldots, l)$ according to the solution of the assignment problem (9)-(12).

From Lemma 1, the first step can obtain the optimal group schedule of minimizing $\sum_{i=1}^{l}\sum_{t=1}^{l} s_i^t x_{it}$ in $O(l^3)$ time. From Lemma 2, the second step can obtain the optimal schedule of the jobs in group G_i to minimize $\sum_{j=1}^{n_i}\sum_{r=1}^{n_i} p_{ij}^r x_{jr}$ in $O(n_i^3)$ time. Then, the total computational complexity of the two steps is $O(l^3) + \sum_{j=1}^{l} O(n_i^3)$. Since $n = \sum_{j=1}^{l} n_i$, it can be obtained that $l^3 \leqslant n^3$ and $\sum_{j=1}^{l} n_i^3 \leqslant n^3$. Hence, the upper bound of the total computational complexity of optimally solving the problem $1 \mid \mathrm{PDG}, \mathrm{PDP} \mid C_{\max}$ is $O(n^3)$.

In the following, we consider a special case of the PDG model. The actual setup time of group G_i scheduled in the t-th position in group sequence is defined by

$$s_i^t = f_s(s_i)\, g_s(r) \qquad t = 1, 2, \ldots, l \tag{13}$$

where $f_s(s_i)$ is the function of s_i, and $g_s(t)$ is the function of t. While two groups are scheduled in the same position, the group with a longer normal setup time always has a longer actual setup time. In this paper, $f_s(s_i)$ is assumed to be non-decreasing in s_i. For simplicity, we use SPDG to denote this kind of special position-dependent group setup time model.

Lemma 3 In the SPDG model, if $g_s(r)$ is non-decreasing in r, $\sum_{i=1}^{l}\sum_{t=1}^{l} s_i^t x_{it}$ can be minimized by sequencing all the groups in the classic shortest processing time first (SPT) rule about the normal group setup times; otherwise, if $g_s(r)$ is non-increasing in r, $\sum_{i=1}^{l}\sum_{t=1}^{l} s_i^t x_{it}$ can be minimized by sequencing all the groups in the classic longest processing time first (LPT) rule about the normal group setup times.

Proof The results can be obtained by a standard pair-wise interchange technique.

Considering a special case of the PDP model, the actual processing time of job J_{ij} scheduled in the r-th position in group G_i is denoted as

$$p_{ij}^r = f_p(p_{ij})\, g_p(r) \qquad r = 1, 2, \ldots, n_i \tag{14}$$

where $f_p(p_{ij})$ and $g_p(r)$ are the functions of p_{ij} and r, respectively. Similar to the function $f_s(s_i)$, the actual setup time function $f_p(p_{ij})$ is assumed to be non-decreasing in p_{ij}. We use SPDP to denote this special position-dependent processing time model.

Lemma 4 In the SPDP model, if $g_p(r)$ is non-decreasing in r in group $G_i (i = 1, 2, \ldots, l)$, $\sum_{r=1}^{n_i} p_{ij}^r x_{jr}$ can be minimized by sequencing all the groups in the SPT rule, else if $g_p(r)$ is non-increasing in r, $\sum_{r=1}^{n_i} p_{ij}^r x_{jr}$ can be minimized by sequencing all the groups in the LPT rule.

Proof Similar to the proof of Lemma 1, the results of this lemma can also be obtained by a standard pair-wise interchange technique.

Theorem 2 1) The $1 \mid \mathrm{PDG}, \mathrm{SPDP} \mid C_{\max}$ problem can be optimally solved in $O(l^3) + O(n\log n)$ time;

2) The $1 \mid \mathrm{SPDG}, \mathrm{PDP} \mid C_{\max}$ problem can be optimally solved in $O(n^3)$ time;

3) The $1 \mid \mathrm{SPDG}, \mathrm{SPDP} \mid C_{\max}$ problem can be optimally solved in $O(n\log n)$ time.

Proof With the similar analysis method in Theorem 1, it can be analyzed that the problems $1 \mid \mathrm{PDG}, \mathrm{SPDP} \mid C_{\max}$, $1 \mid \mathrm{SPDG}, \mathrm{PDP} \mid C_{\max}$ and $1 \mid \mathrm{SPDG}, \mathrm{SPDP} \mid C_{\max}$ can be optimally solved in $O(l^3) + O(n\log n)$, $O(n^3)$ and $O(n\log n)$ time, respectively.

In the following, we consider the $P_m \mid \mathrm{PDG}, \mathrm{PDP} \mid L_{\mathrm{T}}$ problem. We first propose two lemmas, which are useful in the following theorem. Note that

$$L_T = \sum_{i=1}^{l}\sum_{k=1}^{m}\sum_{t=1}^{l} s_{ik}^t x_{ikt} + \sum_{i=1}^{l}\sum_{j=1}^{n_i}\sum_{r=1}^{n_i} p_{ij}^r x_{jr}$$

Since the term of $\sum_{i=1}^{l}\sum_{j=1}^{n_i}\sum_{r=1}^{n_i} p_{ij}^r x_{jr}$ in L_T is the same as the second part of makespan (i. e., Eq. (3)), we focus on the $\sum_{i=1}^{l}\sum_{k=1}^{m}\sum_{t=1}^{l} s_{ik}^t x_{ikt}$ minimization scheduling on parallel machines.

Lemma 5 For the m parallel machines scheduling, the $\sum_{i=1}^{l}\sum_{k=1}^{m}\sum_{t=1}^{l} s_{ik}^t x_{ikt}$ minimization problem can be optimally solved in $O(l^{m+2})$ time.

Proof Each group and its setup time can be viewed as a special job and its processing time, respectively. Then, the scheduling of each group can be sorted by the method in Ref. [14]. Similar to the analysis in Ref. [14], it is easy to obtain that the total processing time minimization problem can be optimally solved in $O(l^{m+2})$ time.

Lemma 6 For the m parallel machines scheduling, if the actual group setup times are increasing in position r, the $\sum_{i=1}^{l}\sum_{k=1}^{m}\sum_{t=1}^{l} s_{ik}^t x_{ikt}$ minimization problem can be optimally solved in $O(l^3)$ time.

Proof In the following, we will prove this lemma in three steps.

First, we will show that all the groups should be scheduled continuously from the starting of each machine. Let l_i denote the number of groups on machine i. If a group is assigned to position r on machine i, but no group is assigned to position $r-1$ on this machine. Due to the increasing monotonicity of the group setup time, the $\sum_{i=1}^{l}\sum_{k=1}^{m}\sum_{t=1}^{l} s_{ik}^t x_{ikt}$ can be reduced by moving the job from position r to position $r-1$ while the other groups are unchanged. By repeating this procedure, we guarantee that all the groups should be scheduled continuously from the starting of each machine in a feasible schedule.

Secondly, we prove that the number of groups on each machine should follow the machine balance principle (i. e., $\lfloor l/m \rfloor \leqslant l_i \leqslant \lfloor l/m \rfloor + 1$). The group, machine and objective $\sum_{i=1}^{l}\sum_{k=1}^{m}\sum_{t=1}^{l} s_{ik}^t x_{ikt}$ are quite similar to the job, group and makespan in Ref. [17], respectively. A contradiction method was used to prove the group balance principle in Ref. [17], and it is easy to use the similar method in Ref. [17] to prove the machine balance principle for this problem.

Finally, since all the groups should be scheduled continuously and the number of groups on each machine should follow the machine balance principle, we can obtain the group allocation on each machine (i. e., the first $\lfloor l/m \rfloor \leqslant l_i \leqslant \lfloor l/m \rfloor + 1$ positions on each parallel machine). That is to say, each of the l jobs need not to be potentially assigned to any position on any of the machines (i. e., to lm positions), but to l positions (i. e., $\sum_{i=1}^{m} l_i = l$). Therefore, the problem of minimizing $\sum_{i=1}^{l}\sum_{k=1}^{m}\sum_{t=1}^{l} s_{ik}^t x_{ikt}$ can be transferred as the following assignment problem with $l \times l$ group positions:

$$\min \sum_{i=1}^{l}\sum_{k=1}^{m}\sum_{t=1}^{l_k} s_{ik}^t x_{ikt} \tag{15}$$

s. t.

$$\sum_{i=1}^{l} x_{it} = 1 \qquad i = 1, 2, \dots, l \tag{16}$$

$$\sum_{k=1}^{l}\sum_{t=1}^{l_k} x_{ikt} = 1 \qquad k = 1, 2, \dots m;\ t = 1, 2, \dots, l_k \tag{17}$$

$$x_{ikt} = 1 \text{ or } 0 \qquad i = 1, 2, \dots, l;\ k = 1, 2, \dots, m;\ t = 1, 2, \dots, l_k \tag{18}$$

where $x_{ikt} = 1$ if group G_i is scheduled in the t-th position on machine k, and $x_{ikt} = 0$, otherwise. The above assignment problem (15)-(18) is an $l \times l$ group-position problem, which can be solved in $O(l^3)$ time (by the Hungarian method, see, e. g. Ref. [16]).

Let INC denote that the actual group setup time s_{ik}^t is increasing in t. Then, the parallel machines group scheduling problem with increasing position-dependent group setup times and total load minimization objective can be denoted as $P_m \mid \text{INC, PDG, PDP} \mid L_T$. We can denote the other relevant problems by the same method.

Theorem 3 1) The problem $P_m \mid \text{PDG, PDP} \mid L_T$ can be optimally solved in $O(l^{m+2}) + O(n^3)$) time;

2) The problem $P_m \mid \text{PDG, SPDP} \mid L_T$ can be optimally solved in $O(l^{m+2}) + O(n\log n)$ time;

3) The problem $P_m \mid \text{INC, PDG, PDP} \mid L_T$ can be optimally solved in $O(l^3) + O(n^3)$) time;

4) The problem $P_m \mid \text{INC, PDG, SPDP} \mid L_T$ can be optimally solved in $O(l^{m+2}) + O(n\log n)$ time.

Proof Based on the conclusions of Lemmas 1 to 6, it is easy to prove this theorem by the similar proof method of Theorem 1.

In the following, we list the computational complexity of all the considered problems in Tab. 1.

Based on the above results, we propose a solving algorithm for the group scheduling problems with general position-dependent effects.

Algorithm 1

Step 1 Based on the results of Lemmas 1, 3, 5 and 6, schedule all the groups.

If all the actual group setup times satisfy the SPDG model in single machine scheduling, list all the groups by the LPT rule (i. e., increasing case) or the SPT rule (i. e., non-increasing case). If all the actual group setup

Tab. 1 Computational complexity of several relevant problems

Problem	Complexity	Theorem
$1 \mid \text{PDG, PDP} \mid C_{\max}$	$O(n^3)$	Theorem 1
$1 \mid \text{PDG, SPDP} \mid C_{\max}$	$O(l^3)+O(n\log n)$	Theorem 2
$1 \mid \text{SPDG, PDP} \mid C_{\max}$	$O(n^3)$	Theorem 2
$1 \mid \text{SPDG, SPDP} \mid C_{\max}$	$O(n\log n)$	Theorem 2
$P_m \mid \text{PDG, PDP} \mid L_{\mathrm{T}}$	$O(l^{m+2})+O(n^3)$	Theorem 3
$P_m \mid \text{PDG, SPDP} \mid L_{\mathrm{T}}$	$O(l^{m+2})+O(n\log n)$	Theorem 3
$P_m \mid \text{INC, PDG, PDP} \mid L_{\mathrm{T}}$	$O(l^3)+O(n^3)$	Theorem 3
$P_m \mid \text{INC, PDG, SPDP} \mid L_{\mathrm{T}}$	$O(l^3)+O(n\log n)$	Theorem 3

times satisfy the PDG model in single machine scheduling, list all the groups by the classic Hungarian algorithm. If all the group setup times simultaneously satisfy the INC model and the PDG model in parallel machines scheduling, schedule all the groups by the classic Hungarian algorithm and the machine balance principle. If all the actual group setup times satisfy the PDG model in parallel machines scheduling, list all the possible group combinations on the machines, then schedule all the groups by the classic Hungarian algorithm for each group combination.

Step 2 Based on the results of Lemmas 2 and 4, schedule all the jobs in groups.

If all the job processing times satisfy the SPDG model, list all the jobs by the LPT rule (i. e., increasing case) or the SPT rule (i. e, non-increasing case). If all the job processing setup times satisfy the PDG model, list all the jobs by the classic Hungarian algorithm.

Step 3 Insert all the job sequences in the corresponding group, then we obtain the total schedule.

Example 1 The following example is given to illustrate the efficiency of Algorithm 1. The algorithms were coded in Matlab version 2010a and the experiments were performed on a personal computer powered by an Intel Pentium(R) Dual-Core CPU E6300 @ 2. 80 GHz with 512 MB RAM operating under Windows XP. Consider an example that there are three parallel machines in a workshop and 30 jobs, which are divided into nine groups and need to be scheduled on the machines. All the actual group setup times on parallel machines are given in Tab. 2. We propose the actual job processing times of all the jobs in Tabs. 3 to 7. From Tabs. 2 to 7, it can be seen that all the group setup times simultaneously satisfy the INC model and the PDG model, and the actual job processing times satisfy the PDG model. Based on Algorithm 1, we list all the groups by the classic Hungarian algorithm and the machine balance principle, and schedule all the jobs by the classic Hungarian algorithm. The computational results show that the optimal total load (i. e., 153 min) is obtained in 0. 950 4 s. The optimal schedule is proposed in Tab. 8.

Tab. 2 Actual group setup times

Job number	Position number								
	1	2	3	4	5	6	7	8	9
1	2	3	8	10	14	14	16	17	17
2	2	5	7	9	10	12	14	18	20
3	4	11	11	13	14	19	20	20	20
4	1	1	7	7	11	15	15	16	19
5	1	3	4	6	9	12	16	17	19
6	3	3	5	6	6	6	8	16	19
7	1	6	12	13	14	14	16	16	16
8	2	6	8	10	10	14	16	16	17
9	1	4	4	6	8	11	12	15	17

Tab. 3 Job processing times of groups 1, 3

Job number	Position number		
	1	2	3
1	10	10	2
2	2	5	5
3	10	9	10

Tab. 4 Job processing times of groups 2, 6

Job number	Position number		
	1	2	3
1	8	1	7
2	10	9	8
3	7	10	8

Tab. 5 Job processing times of groups 4, 5

Job number	Position number		
	1	2	3
1	4	8	1
2	7	1	1
3	2	3	9

Tab. 6 Job processing times of groups 7, 8

Job number	Position number			
	1	2	3	4
1	5	8	2	4
2	5	3	2	6
3	7	7	5	3
4	8	7	10	8

Tab. 7 Job processing times of group 9

Job number	Position number			
	1	2	3	4
1	3	10	3	3
2	6	6	9	10
3	7	2	3	4
4	9	2	9	2

Tab. 8 Optimal schedule and results

Machine	Sequence on the machine
Machine 1	$G_9(J_{91}, J_{94}, J_{92}, J_{93})$, $G_6(J_{63}, J_{61}, J_{62})$, $G_4(J_{43}, J_{42}, J_{41})$
Machine 2	$G_8(J_{84}, J_{82}, J_{81}, J_{83})$, $G_2(J_{23}, J_{21}, J_{22})$, $G_5(J_{53}, J_{52}, J_{51})$
Machine 3	$G_7(J_{74}, J_{72}, J_{71}, J_{73})$, $G_1(J_{12}, J_{13}, J_{11})$, $G_3(J_{32}, J_{33}, J_{31})$

3 Conclusion

In this paper, we investigate group scheduling with general position-dependent group setup times and job processing times. The single machine makespan minimization problem and the parallel-machine total load minimization problem are both considered. Moreover, several special cases, where the group setup time or the job processing time is described by a product function, are also investigated. From our studies, it can be seen that all the studied problems are polynomially solvable. The time complexity results for all the considered problems are presented. Our future research will be directed to investigate other models of general learning or aging effects on group scheduling, or other performance measures.

References

[1] Cheng T C E, Ding Q, Lin B M T. A concise survey of scheduling with time-dependent processing times [J]. *European Journal of Operational Research*, 2004, **152**(1): 1 − 13.

[2] Biskup D. A state-of-the-art review on scheduling with learning effects [J]. *European Journal of Operational Research*, 2008, **188**(2): 315 − 329.

[3] Janiak A, Rudek R. Experience based approach to scheduling problems with the learning effect [J]. *IEEE Transactions on Systems, Man, and Cybernetics—Part A*, 2009, **39**(2): 344 − 357.

[4] Yin Y Q, Xu D H, Sun K B, et al. Some scheduling problems with general position-dependent and time-dependent learning effects [J]. *Information Science*, 2009, **179**(14): 2416 − 2425.

[5] Ng C T, Wang J B, Cheng T C E, et al. Two-machine flow shop scheduling with deteriorating jobs to minimize total completion time [J]. *Computers and Operations Research*, 2010, **37**(1): 83 − 90.

[6] Ham I, Hitomi K, Yoshida T. *Group technology: applications to production management* [M]. Boston: Kluwer-Nijhoff, 1985.

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

[8] Yang S J, Yang D L. Single-machine group scheduling problems under the effects of deterioration and learning [J]. *Computers and Industrial Engineering*, 2010, **58**(4): 754 − 758.

[9] Liu P, Tang L X, Zhou X Y. Two-agent group scheduling with deteriorating jobs on a single machine [J]. *International Journal of Advanced Manufacturing Technology*, 2010, **47**(5): 657 − 664.

[10] Wang J B, Sun L Y. Single-machine group scheduling with linearly decreasing time-dependent setup times and job processing times [J]. *International Journal of Advanced Manufacturing Technology*, 2010, **49**(5): 765 − 772.

[11] Lee W C, Wu C C. A note on single-machine group scheduling problems with position-based learning effect [J]. *Applied Mathematical Modelling*, 2009, **33**(4): 2159 − 2163.

[12] Zhang X G, Yan G L. Single-machine group scheduling problems with deteriorated and learning effect [J]. *Applied Mathematics and Computation*, 2010, **216**(4): 1259 − 1266.

[13] Wang J B, Wang M Z. A revision of machine scheduling problems with a general learning effect [J]. *Mathematical and Computer Modelling*, 2011, **53**(1): 330 − 336.

[14] Mosheiov G. Proportionate flowshops with general position-dependent processing times [J]. *Information Processing Letters*, 2011, **111**(4): 174 − 177.

[15] Graham R L, Lawler E L, Lenstra J K, et al. Optimization and approximation in deterministic sequencing and scheduling: a survey [J]. *Annals of Discrete Mathematics*, 1979, **5**: 287 − 326.

[16] Kuhn H W. The Hungarian method for the assignment problem [J]. *Naval Research Logistics Quarterly*, 1955, **2**: 83 − 97.

[17] Zhao C L, Tang H Y. Single machine scheduling with general job-dependent aging effect and maintenance activities to minimize makespan [J]. *Applied Mathematical Modelling*, 2010, **34**(3): 837 − 841.

带有一般性位置依赖影响的分组调度研究

虞先玉[1,2]　张玉林[1]　张玉环[1]

([1]东南大学经济管理学院,南京 211189)
([2]东华理工大学理学院,南昌 330000)

摘要:为了研究更具实际意义的带有位置依赖影响的分组调度决策问题,建立了一般性位置依赖的分组调度模型. 在模型中,分组实际发动时间和工件的实际加工时间被表示成初始时间和调度位置的一般函数. 此类函数没有被假设为特殊函数形式,且没有要求限制其函数单调性. 通过数理逻辑分析和证明,把所研究的问题模型分解为组调度过程和工件调度过程,并把每个调度过程分别转化为经典任务分派问题和单机排序调度问题,进而分析问题求解的计算复杂度. 研究表明,即使在一般性位置依赖的模型假设下,单机最小化时间表长的分组调度问题和平行机最小化总负荷的分组调度问题仍然是多项式可解的.

关键词:分组调度;位置依赖;时间表长;总负荷

中图分类号:F406.2; TP301