A note on resource allocation scheduling with group technology and learning effect on a single machine

Yuan-Yuan Lu[a,1], Ji-Bo Wang[b], Ping Ji[c], Hongyu He[d]

[a]College of Mathematics, Jilin Normal University, Siping, Jilin 136000, China;

[b]School of Science, Shenyang Aerospace University, Shenyang 110136, China;

[c]Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China;

[d]Department of Industrial Engineering, Management College of Engineering, Peking University, Beijing 100871, China

Abstract

In this paper, a single machine group scheduling with learning effect and convex resource allocation is studied. The goal is to find the optimal job schedule, the optimal group schedule, and resource allocations of jobs and groups. For the problem of minimizing the makespan subject to limited resource availability, it is proved that the problem can be solved in polynomial time under the condition that the setup time of groups are independent. For the general setup time of groups, a heuristic algorithm and a branch-and-bound algorithm are proposed respectively. The computational experiments show that the performance of the heuristic algorithm is fairly accurately in obtaining near-optimal solutions.

Keywords: Scheduling; Learning effect; Resource allocation; Group technology; Heuristic algorithm; Branch-and-bound algorithm

1 Introduction

In recent years, a lot of work has been done on production systems in which job processing times may be changing due to the phenomenon of learning and/or resource allocation. Extensive surveys of different scheduling models and problems involving learning and resource allocation (controllable processing times) can be found in Biskup (2008) and Shabtay and Steiner (2007). More recently, papers Niu et al. (2015), Shiau et al. (2015), Zhang and Wang (2015), and Xu et al. (2016) considered scheduling problems with learning effects. Niu et al. (2015) considered some scheduling problems in which the actual processing time of a job is a function of the sum of the function of the processing times of the jobs already processed and job position. They proved that some single machine scheduling problems and some special cases of flow shop scheduling problems can be solved in polynomial time. Shiau et al. (2015) considered a two-machine flow-shop scheduling with learning effects. For a two-agent case (i.e., to minimize the total completion

[1] Corresponding author. E-mail addresses: luyuanyuan_jilin@163.com

time of the jobs from one agent, given that the maximum tardiness of the jobs from the other agent cannot exceed a bound), they proposed a branch-and-bound algorithm and several genetic algorithms. Zhang and Wang (2015) considered single-machine due window assignment scheduling with learning effect. Xu et al. (2016) considered an order scheduling with position-based learning effect. Papers Yang et al. (2014), Hsieh et al. (2015), Jiang et al. (2015), Herr and Goel (2016) considered scheduling problems with resource allocation (controllable processing times). Yang et al. (2014) considered single machine resource allocation scheduling problems with multiple due windows assignment. They proved that the problems can be solved in polynomial time respectively. Hsieh et al. (2015) considered unrelated parallel machine scheduling with discrete controllable processing times. Jiang et al. (2015) considered a hybrid flow shop scheduling problem with controllable processing times. Herr and Goel (2016) considered total tardiness minimization single machine scheduling problem with family setups and resource constraints. Papers Liu and Feng (2014), Lu et al. (2014), Wang and Wang (2014b), Yin et al. (2014), Li et al. (2015), Wang and Wang (2015), and He et al. (2016) considered scheduling problems with simultaneous considerations of learning effects and resource allocation. Liu and Feng (2014) considered flowshop scheduling with learning effect and convex resource-dependent processing times. For the two-machine no-wait case, they proved two cost functions can be solved in polynomial time. Lu et al. (2014) considered two due-date assignment problems with learning effect and resource-dependent processing times. Wang and Wang (2014b) considered common due-window assignment problems with learning effect and resource-dependent processing times. Li et al. (2015) considered slack due-window assignment problems with learning effect and resource-dependent processing times. Wang and Wang (2015) considered scheduling problems with job-dependent learning effect and convex resource-dependent processing times. He et al. (2016) considered scheduling problems with general truncated job-dependent learning effect and resource-dependent processing times.

On the other hand, the production efficiency can be increased by group technology, i.e., grouping various parts and products with similar designs and or production processes (Shabtay et al. (2010), Wang and Wang (2014a), Yin et al. (2014), Keshavarz et al. (2015)). Qin et al. (2016) considered flowshop scheduling with group technology and position-based learning effect. To the best of our knowledge, apart from the recent paper of Zhu et al. (2011), it has not been investigated the scheduling problems with simultaneous considerations of learning effects, resource allocation and group technology. *"The phenomena of learning effects, resource allocation and group technology occurring simultaneously can be found in many real-life situations. For example, in the chemical industry, the processing time of a chemical compound can be changed by increasing the amount of catalysts, which entails some extra costs (Wang and Cheng [2005]). Clearly, compressing jobs would be rational and possible only if the additional cost is* *compensated by the gains from job completion at an earlier time. The scheduling problem with controllable processing times is concerned with determining not only the job sequence but also the amount of compression for each job so as to minimize the total cost. On the other hand, the learning effects reflect that the workers become more skilled to operate the machines through experience accumulation, and group technology can increase the production efficiency. For this situation, considering these the job learning effects, resource allocation and group technology in job scheduling is both necessary and reasonable. (Wang et al. [2010], and Zhu et al. [2011])."* Zhu et al. (2011) considered single-machine group scheduling with resource allocation and learning effect, i.e., the linear resource allocation model: $p_{ij}^A = p_{ij}r^{b_1}l^{b_2} - \beta_{ij}u_{ij}$, the convex resource allocation model: $p_{ij}^A = \left(\frac{p_{ij}r^{b_1}l^{b_2}}{u_{ij}}\right)^k$, $u_{ij} > 0$, and $s_i^A = s_i r^{b_1}$, where $b_1 \leq 0$ and $b_2 \leq 0$ denote the learning indices of the group and job learning effect respectively, and u_{ij} is the amount of a non-renewable resource allocated to job J_{ij}, with $0 \leq u_{ij} \leq \bar{u}_{ij} < \frac{p_{ij}m^{b_1}(n_m)^{b_2}}{\beta_{ij}}$, where $\bar{u}_{ij}$ denote the maximum amount of resource allocated to job J_{ij} and β_{ij} is the positive compression rate of job J_{ij}, $k > 0$ is a given constant. They showed that the problems for minimizing the weighted sum of makespan and total resource cost remain polynomially solvable. In the real production management, the resource is precious, and the resource availability is limited, hence in this study, the results of Zhu et al. (2011) are extended, that is, the single machine group scheduling with learning effect and resource allocation is considered subject to limited resource availability .

The remainder of this article is organized as follows. Section 2 formulates the problem. Section 3 derives the properties of the optimal schedule for the problem and provide solution algorithms for them. Section 4 provides some numerical experiments. Section 5 gives some extensions. The last section contains some conclusions and suggests some future research topics.

2 Problem formulation

The model is described as follows. n independent jobs to be grouped into m groups, and to be processed on a single machine is considered. All jobs and the machine are available at time zero. It is assumed that there is no setup time between two consecutive jobs in the same group. However, a group setup time is required to process a group. It is also assumed that the processing of a job may not be interrupted. Let n_i be the number of jobs belonging to group G_i, thus, $n_1 + n_2 + \ldots + n_m = n$, and J_{ij} denote the jth job in group G_i, $i = 1, 2, \ldots, m; j = 1, 2, \ldots, n_i$. Let p_{ij} be the original (normal) processing time of job J_{ij}. As in Zhu et al. (2011), if the job J_{ij} is scheduled in group position r and internal job position l, the actual processing time of the job is

$$p_{ij}^{A} = \left(\frac{p_{ij} r^{a_1} l^{a_2}}{u_{ij}}\right)^{k}, u_{ij} > 0, \tag{1}$$

where $k > 0$ is a given constant, $a_1 \leq 0$ and $a_2 \leq 0$ denote the learning indices of the group and job learning effect respectively. As in Zhu et al. (2011), the actual setup time of group G_i, if it is scheduled in the rth group in the sequence, is given by

$$s_i^A = s_i r^{a_3}, \tag{2}$$

where s_i is the original (normal) setup time of group G_i, $a_3 \leq 0$ denotes the learning index of the group learning effect.

For a given schedule S, let $C_{ij} = C_{ij}(S)$ denote the completion time of job J_{ij} in group G_i under schedule S, and $C_{\max} = \max\{C_{ij} | i = 1, 2, \ldots, m; j = 1, 2, \ldots, n_i\}$ denote the makespan of a given schedule. Our goal is to find the job sequence π_i^* $(i = 1, 2, \ldots, m)$, the optimal group sequence ζ^*, and the resource allocation u^* so as to minimize $C_{\max}$ subject to $\sum_{i=1}^{m} \sum_{j=1}^{n_i} u_{ij} \leq U$, where $U > 0$ is a given real number and denotes the total resource amount spent on compression. Using the three-field notation (Graham et al. 1979), the problem can be denoted as $1 \left| GT, CRALE, \sum_{i=1}^{m} \sum_{j=1}^{n_i} u_{ij} \leq U \right| C_{\max}$, where $CRALE$ means "convex resource allocation and learning effect". Zhu et al. (2011) considered the problem $1 \left| GT, CRALE, a_3 = a_1 \right| \beta_1 C_{\max} + \beta_2 \sum_{i=1}^{m} \sum_{j=1}^{n_i} u_{ij}$, where β_1 and β_2 are positive parameters decided by the decision-makers, they proved that the problem $1 \left| GT, CRALE, a_3 = a_1 \right| \beta_1 C_{\max} + \beta_2 \sum_{i=1}^{m} \sum_{j=1}^{n_i} u_{ij}$ cab be solved in polynomial time.

3 Main results

Obviously, for the $1 \left| GT, CRALE, \sum_{i=1}^{m} \sum_{j=1}^{n_i} u_{ij} \leq U \right| C_{\max}$ problem, the optimal resource constraint is satisfied as equality, i.e., $\sum_{i=1}^{m} \sum_{j=1}^{n_i} u_{ij} = U$.

Lemma 1 *For the* $1 \left| GT, CRALE, \sum_{i=1}^{m} \sum_{j=1}^{n_i} u_{ij} \leq U \right| C_{\max}$ *problem, the optimal resource allocation can be determined as follows:*

$$u_{[i][j]}^{*} = \frac{\left(p_{[i][j]} i^{a_1} j^{a_2}\right)^{\frac{k}{k+1}}}{\sum_{i=1}^{m} \sum_{j=1}^{n_{[i]}} \left(p_{[i][j]} i^{a_1} j^{a_2}\right)^{\frac{k}{k+1}}} \times U \tag{3}$$

for $i = 1, 2, \ldots, m; j = 1, 2, \ldots, n_i$.

Proof For a given sequence, the Lagrange function is

$$L(\mathbf{u}, \lambda) = C_{\max} + \lambda \left(\sum_{i=1}^{m} \sum_{j=1}^{n_{[i]}} u_{[i][j]} - U \right)$$

$$= \sum_{i=1}^{m} s_{[i]} i^{a_3} + \sum_{i=1}^{m}\sum_{j=1}^{n_{[i]}} \left(\frac{p_{[i][j]} i^{a_1} j^{a_2}}{u_{[i][j]}}\right)^k + \lambda \left(\sum_{i=1}^{m}\sum_{j=1}^{n_{[i]}} u_{[i][j]} - U\right) \tag{4}$$

where λ is the Lagrangian multiplier. Deriving (4) with respect to $u_{[i][j]}$ and λ, it has

$$\frac{\partial L(\mathbf{u},\lambda)}{\partial u_{[i][j]}} = \lambda - k \times \frac{\left(p_{[i][j]} i^{a_1} j^{a_2}\right)^k}{(u_{[i][j]})^{k+1}} = 0,\ \forall i = 1,2,\ldots,m. \tag{5}$$

$$\frac{\partial L(\mathbf{u},\lambda)}{\partial \lambda} = \sum_{i=1}^{m}\sum_{j=1}^{n_{[i]}} u_{[i][j]} - U = 0 \tag{6}$$

Using (5) and (6), it has

$$u_{[i][j]} = \frac{\left(k\left(p_{[i][j]} i^{a_1} j^{a_2}\right)^k\right)^{1/(k+1)}}{\lambda^{1/(k+1)}} \tag{7}$$

and

$$\lambda^{1/(k+1)} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_{[i]}} (k)^{1/(k+1)} \left(p_{[i][j]} i^{a_1} j^{a_2}\right)^{k/(k+1)}}{U}. \tag{8}$$

From (7) and (8), for $i = 1,2,\ldots,m$, it has

$$u^*_{[i][j]} = \frac{\left(p_{[i][j]} i^{a_1} j^{a_2}\right)^{\frac{k}{k+1}}}{\sum_{i=1}^{m}\sum_{j=1}^{n_{[i]}} \left(p_{[i][j]} i^{a_1} j^{a_2}\right)^{\frac{k}{k+1}}} \times U.$$

□

Using Lemma 1 and substituting (3) into $C_{\max}$, it has

$$\begin{aligned} C_{\max} &= \sum_{i=1}^{m} s_{[i]} i^{a_3} + \sum_{i=1}^{m}\sum_{j=1}^{n_{[i]}} \left(\frac{p_{[i][j]} i^{a_1} j^{a_2}}{u_{[i][j]}}\right)^k \\ &= \sum_{i=1}^{m} s_{[i]} i^{a_3} + U^{-k} \left(\sum_{i=1}^{m}\sum_{j=1}^{n_{[i]}} \left(p_{[i][j]} i^{a_1} j^{a_2}\right)^{\frac{k}{k+1}}\right)^{k+1} \\ &= \sum_{i=1}^{m} s_{[i]} i^{a_3} + U^{-k} \left(\sum_{i=1}^{m}\sum_{j=1}^{n_{[i]}} \mu_{[i][j]} \omega_{[i]j}\right)^{k+1}, \end{aligned} \tag{9}$$

where $\mu_{[i][j]} = \left(p_{[i][j]}\right)^{\frac{k}{k+1}}$ and $\omega_{[i]j} = (i^{a_1} j^{a_2})^{\frac{k}{k+1}}$.

The value $i^{\frac{ka_1}{k+1}}$ is identical for any job $J_{[i]j}$ within the group $G_{[i]}$ of any possible position in ζ. So

$$\omega_{ij} = j^{\frac{ka_2}{k+1}}, i = 1,2,\ldots,m; j = 1,2,\ldots,n_i, \tag{10}$$

can be used to calculate the position weight of jobs in group G_i.

Lemma 2 *For the* $1\left|GT, CRALE, \sum_{i=1}^{m}\sum_{j=1}^{n_i} u_{ij} \le U\right|C_{\max}$ *problem, the optimal job sequence* π_i^* *within each group can be obtained by sequencing the jobs in non-decreasing order of* p_{ij} *(the the smallest processing time (SPT) first rule).*

Proof For a given group G_i $(i = 1, 2, \ldots, m)$, from (10), $\omega_{ij} = j^{\frac{ka_2}{k+1}}$ $(j = 1, 2, \ldots, n_i)$ is a decreasing function on j $(a_2 \le 0)$, hence by the HLP rule (Hardy et al. (1967)), the result can be easily obtained. □

3.1 Case $a_3 = 0$

For the $1\left|GT, CRALE, \sum_{i=1}^{m}\sum_{j=1}^{n_i} u_{ij} \le U\right|C_{\max}$ problem, the complexity of determining the optimal group sequence is still remains an open question, so the special case $a_3 = 0$ (i.e., $s_i^A = s_i$) should be considered.

Lemma 3 *For the* $1\left|GT, CRALE, \sum_{i=1}^{m}\sum_{j=1}^{n_i} u_{ij} \le U\right|C_{\max}$ *problem, the optimal group sequence* ζ^* *under the condition of* $a_3 = 0$*, can be obtained by solving a Linear Assignment Problem.*

Proof It is noted that the optimal sequence π_i^* of group $G_{[i]}$ can be predetermined by Lemma 2, and the value $\sum_{i=1}^{m} s_{[i]} i^{a_3} = \sum_{i=1}^{m} s_i$ is a constant number. Now it should be considered that how to determine the optimal group sequence. It should be defined that the binary variables x_{ir} such that $x_{ir} = 1$ if group G_i is assigned to position r and $x_{ir} = 0$ otherwise, $i = 1, 2, \cdots, m$ and $r = 1, 2, \cdots, m$.

Let

$$A_{ir} = \sum_{j=1}^{n_i} \left(p_{i[j]} r^{a_1} j^{a_2}\right)^{\frac{k}{k+1}}, \qquad i, r = 1, 2, \cdots, m. \tag{11}$$

Consequently, the optimal group sequence determining problem can be formulated as the following Assignment Problem:

$$\textbf{AP} \quad \min \sum_{i=1}^{m}\sum_{r=1}^{m} A_{ir} x_{ir} \tag{12}$$

subject to

$$\sum_{r=1}^{m} x_{ir} = 1, \quad i = 1, 2, \ldots, m, \tag{13}$$

$$\sum_{i=1}^{m} x_{ir} = 1, \quad r = 1, 2, \ldots, m, \tag{14}$$

$$x_{ir} = 0 \text{ or } 1, \quad i, r = 1, 2, \ldots, m. \tag{15}$$

□

Based on the above analysis, the problem $1\left|GT, CRALE, \sum_{i=1}^{m}\sum_{j=1}^{n_i} u_{ij} \le U, a_3 = 0\right| C_{\max}$ can be optimally solved by the following algorithm.

Algorithm 1

Step 1. Determine the initial internal job sequence π_i^* by using the SPT rule (Lemma 2).

Step 2. Calculate all A_{ir} by Equation (11) with π_i^* for $i = 1, 2, \ldots, m$ and $r = 1, 2, \ldots, m$.

Step 3. Solve the Assignment Problem **AP** to determine the optimal group sequence ζ^*.

Step 4. Calculate the optimal resources allocation $u^*_{[i][j]}(\zeta^*, \pi_1^*, \pi_2^*, \ldots, \pi_m^*)$ by using Equation (3).

Theorem 1 *Algorithm 1 solves the problem* $1\left|GT, CRALE, \sum_{i=1}^{m}\sum_{j=1}^{n_i} u_{ij} \le U, a_3 = 0\right| C_{\max}$ *in* $O(n^3)$ *time.*

Proof Similarly to the proof of Zhu et al. (2011). □

Example 1 This example provides an illustration of a 9-job and 3-group problem (i.e., $n = 9$ and $m = 3$). Let $n_1 = 2, n_2 = 3, n_4, a_1 = -0.3, a_2 = -0.25, a_3 = 0$, $k = 2, U = 50$ and the other parameters (i.e., the values p_{ij} and s_i) be shown in Table 1.

Solution.

Step 1. From Lemma 2, it can be obtained that the internal job sequence $\pi_1^* = (J_{11}, J_{12})$, $\pi_2^* = (J_{21}, J_{23}, J_{22})$ and $\pi_3^* = (J_{34}, J_{31}, J_{32}, J_{33})$.

Step 2. The values A_{ir} (by Equation (11)) is given in Table 2.

Step 4. Solve the Assignment Problem **AP**, it can be obtained that the optimal group sequence is $\zeta^* = (G_2, G_1, G_3)$ (see bold in Table 2). The overall schedule is $S^* = (J_{21}, J_{23}, J_{22}; J_{11}, J_{12}; J_{34}, J_{31}, J_{32}, J_{33})$, and $C_{\max} = 6 + 5 + 8 + 7.4736 + 7.8743 + 9.0778 = 43.4257$.

Step 5. From Equation (3), the optimal resources allocation $u_{21}^* = \frac{\left(p_{[i][j]} i^{a_1} j^{a_2}\right)^{\frac{k}{k+1}}}{\sum_{i=1}^{m}\sum_{j=1}^{n_{[i]}}\left(p_{[i][j]} i^{a_1} j^{a_2}\right)^{\frac{k}{k+1}}} \times U = 5.1582$, $u_{23}^* = 5.3325, u_{22}^* = 5.6282; u_{11}^* = 6.5210, u_{12}^* = 8.7776; u_{34}^* = 4.1407, u_{31}^* = 4.2806, u_{32}^* = 4.5180, u_{33}^* = 5.6431$.

Table 1. The data of Example 1

G_i	G_1		G_2			G_3			
J_{ij}	J_{11}	J_{12}	J_{21}	J_{22}	J_{23}	J_{31}	J_{32}	J_{33}	J_{34}
p_{ij}	7	13	4	6	5	5	6	9	4
s_i	6		5			8			

Table 2. The values A_{ir}

	$i\backslash r$	1	2	3
$A_{ir} =$	G_1	8.5849	**7.4736**	6.8914
	G_2	**7.8743**	6.8550	6.3210
	G_3	11.3084	9.8446	**9.0778**

3.2 Case $a_3 \neq 0$

The complexity of the general problem $1\left|GT, CRALE, \sum_{i=1}^{m}\sum_{j=1}^{n_i} u_{ij} \leq U\right|C_{\max}$ remains an open question, it is conjectured that this general problem is NP-hard. From Lemma 2, the optimal job sequence within each group is scheduled by sequencing the jobs in non-decreasing order of p_{ij}. From Section 3.1, a heuristic algorithm for the general problem can be proposed.

Algorithm 2

Step 1. By using the SPT rule (Lemma 2) to determine the internal job sequence π_i^*.
Step 2.1. Calculate all A_{ir} $(i, r = 1, 2, \ldots, m)$ by Equation (11) with π_i^*, and solve the Assignment Problem **AP** to determine the group sequence.
Step 2.2. Groups sequenced in non-decreasing order of s_i;
Step 2.3. Groups sequenced in non-increasing order of $s_i + \sum_{j=1}^{n_i}\left(p_{i[j]}j^{a_2}\right)^{\frac{k}{k+1}}$.
Step 3. From Steps 2.1-2.3, choose the better solution to determine the group sequence ζ.
Step 4. Calculate the optimal resources allocation $u^*_{[i][j]}(\zeta, \pi_1^*, \pi_2^*, \ldots, \pi_m^*)$ by using Equation (3).

In order to solve the general problem $1\left|GT, CRALE, \sum_{i=1}^{m}\sum_{j=1}^{n_i} u_{ij} \leq U\right|C_{\max}$ optimally, a branch-and-bound algorithm is proposed. For the branch-and-bound algorithm, initialization, branching, and bounding are three major components. The initialization solution can be obtained by Algorithm 2, and a depth-first search is adopted in the branching procedure. The efficiency of a branch-and-bound algorithm depends largely on the effectiveness of calculating lower bounds for partial solutions. For each branching node of our problem, it is obvious that if $s_i^A = s_i m^{a_3}$ for unscheduled groups, a minimum makespan can be achieved by Algorithm 1 (this is a lower bound for each branching node).

Management insights: Resource allocation scheduling problems with group technology and learning effect make the process of production decision very difficult. Like the problem considered, the theoretical results can minimize the makespan (cost) and improve the production efficiency.

4 Numerical experiments

Computational experiments are conducted to evaluate the effectiveness of the Algorithm 2 and the branch-and-bound algorithm, and these both algorithms are coded in VC++ 6.0 and ran on CPU Intel® Pentium® 4 2.4 GHz and 2GB RAM. For the experiments, the following parameters are generated from discrete uniform distribution: The original (normal) processing times p_{ij} and the setup times s_i are generated from the uniform distribution over the integers [1, 100], respectively. The learning effects a_1 and a_2 are assigned with uniform distribution over the real numbers [-0.5, -0.1]. Test problems with $U = 100$, $n = 50$ and 100 jobs and with m=10 and 20 families (each group must contain at least one job) are experimented. Moreover, to detect the learning effect a_3 on the performance of an algorithm, three different types of learning rate are assigned, i.e., $a_3 = -0.4, -0.3, -0.2$. For each combination of m, n and a_3, 50 problem instances were randomly generated. A total of 600 problem instances are tested.

For the branch-and-bound algorithm, the average time, and the maximum time (in seconds) are recorded. In order to evaluate the performance of the heuristic algorithm (i.e., Algorithm 2), the average time, and the maximum time (in seconds), the average and maximum error percentage are recorded. The percentage error of the solution produced by the Algorithm 2 is calculated as

$$(C^{HA} - C^*)/C^* \times 100\%,$$

where C^{HA} is the makespan of the solution generated by the **Algorithm 2** and C^* is the makespan of the optimal schedule, which is obtained by a branch-and-bound algorithm. The results are summarized in Table 3. As can be see from Table 3, the the **Algorithm 2** performs very well in terms of the error percentages. The mean error percentage $(C^{HA} - C^*)/C^* \times 100\%$ is less than 1% for all sizes of problems.

The performance of the heuristic algorithm (i.e., Algorithm 2) for n=400 and 1000, m =30 and 60 is also verified by the ratios $(C^{HA} - LB)/LB \times 100\%$ (see Table 4), where LB can be obtained by Algorithm 1 if it is supposed that $s_i^A = s_i m^{a_3}$. As it can be seen from Table 4, the performance of Algorithm 2 is very effective. The average of the ratios $(C^{HA} - LB)/LB \times 100\%$ in Table 4 are less than 1% for all a_3.

5 Extensions

Obviously, the above results can be extended to the following general models:

$$p_{ij}^A = \left(\frac{p_{ij} f(r) g(l)}{u_{ij}}\right)^k, u_{ij} > 0 \tag{16}$$

Table 3 Results for heuristic algorithm and branch-and-bound algorithm

		CPU (s) of branch-and-bound algorithm						CPU (s) of Algorithm 2						Error percentage of Algorithm 2 (%)					
n	m	$a_3=-0.2$		$a_3=-0.3$		$a_4=-0.4$		$a_3=-0.2$		$a_3=-0.3$		$a_4=-0.4$		$a_3=-0.2$		$a_3=-0.3$		$a_4=-0.4$	
		mean	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean	max
	10	74	258	79	267	77	265	1.28	1.45	1.37	1.44	1.37	1.43	0.01	0.51	0.01	0.51	0.01	0.51
50	20	402	2842	405	2867	403	2865	2.87	2.91	2.86	2.89	2.84	2.87	0.12	2.45	0.12	2.44	0.12	2.44
	10	119	388	122	391	115	375	1.99	2.08	1.98	2.07	1.99	2.08	0.01	0.61	0.01	0.63	0.01	0.63
100	20	574	3653	571	3615	569	3601	4.01	4.11	3.99	4.09	4.01	4.11	0.15	2.94	0.15	3.01	0.15	3.03

Table 4 Performance evaluation of Algorithm 2

		CPU (s) of Algorithm 2						Error percentage of Algorithm 2 (%)					
n	m	$a_3 = -0.2$		$a_3 = -0.3$		$a_3 = -0.4$		$a_3 = -0.2$		$a_3 = -0.3$		$a_3 = -0.4$	
		mean	max	mean	max	mean	max	mean	max	mean	max	mean	max
	30	6.13	6.29	6.44	6.60	6.25	6.40	0.25	4.85	0.26	5.09	0.25	4.94
400	60	14.89	15.25	14.18	14.53	14.44	14.80	0.57	11.20	0.60	11.77	0.58	11.41
	30	10.23	10.48	9.74	9.98	9.7845	9.99	0.39	7.70	0.41	8.08	0.39	7.70
1000	60	22.51	23.06	23.63	24.22	22.52	23.08	0.90	17.79	0.95	18.68	0.90	17.80

and

$$s_i^A = s_i h(r), \tag{17}$$

where $f(r)$, $g(l)$ and $h(r)$ are the general non-decreasing and differentiable functions (learning curves), i.e., $1 = f(1) \geq f(2) \geq \ldots \geq f(m)$, $1 = g(1) \geq g(2) \geq \ldots \geq g(n_i)$ and $1 = h(1) \geq h(2) \geq \ldots \geq h(m)$.

6 Conclusions

This paper considered the single machine group scheduling problem with learning effect and resource-dependent processing times. If the setup times of groups have not learning effect, it is showed that the weighted combination of makespan and total resource cost minimization problem can be solved in polynomial time. For the general setup times, a heuristic algorithm and a-branch-and-bound algorithm were given. The algorithms can also be easily applied to the problems with the deterioration (aging) effect (e.g., $a_1 > 0, a_2 > 0$, $a_3 > 0$, $1 = f(1) \leq f(2) \leq \ldots \leq f(m)$, $1 = g(1) \leq g(2) \leq \ldots \leq g(n_i)$ and $1 = h(1) \leq h(2) \leq \ldots \leq h(m)$). For future research, it is worthwhile to study other scheduling problems with deterioration (aging), learning effects, resource allocation and/or group technology (He and Sun [2012, 2015], Rudek and Rudek [2011, 2012]), for example, the job-shop scheduling problems, the parallel machine scheduling problems and other scheduling performance measures.

Acknowledgements—This research was supported by the National Natural Science Foundation of China (grant nos. 71471120 and 71501082).

References

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

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

Hardy, G.H., J.E. Littlewood, and G. Polya. 1967. Inequalities (2nd ed.). Cambridge: Cambridge University Press.

He, H.-Y., M. Liu, and J.-B. Wang. 2016. Resource constrained scheduling with general truncated job-dependent learning effect. Journal of Combinatorial Optimization, DOI:10.1007/s10878-015-9984-5.

He, Y., and L. Sun. 2012. Single-machine group scheduling problems with deterioration to minimize the sum of completion times. Mathematical Problems in Engineering, Volume 2012: Article ID 275239, (9 pages).

He, Y., and L. Sun. 2015. One-machine scheduling problems with deteriorating jobs and position-dependent learning effects under group technology considerations. International Journal of Systems Science 46 (7): 1319-1326.

Herr, H., and A, Goel. 2016. Minimising total tardiness for a single machine scheduling problem with family setups and resource constraints. European Journal of Operational Research 248 (1): 123-135.

Hsieh, P.-H., S.-J. Yang, and D.-L. Yang. 2015. Decision support for unrelated parallel machine scheduling with discrete controllable processing times. Applied Soft Computing 30: 475-483.

Jiang, S., M. Liu, J. Hao, and W. Qian. 2015. A bi-layer optimization approach for a hybrid flow shop scheduling problem involving controllable processing times in the steelmaking industry. Computers & Industrial Engineering 87: 518-531.

Keshavarz, T., M. Savelsbergh, and N. Salmasi. 2015. A branch-and-bound algorithm for the single machine sequence-dependent group scheduling problem with earliness and tardiness penalties. Applied Mathematical Modelling 39 (20): 6410-6424.

Li, G., M.-L. Luo, W.-J. Zhang, and X.-Y. Wang. 2015. Single-machine due-window assignment scheduling based on common flow allowance, learning effect and resource allocation. International Journal of Production Research 53 (4): 1228-1241.

Liu, Y., and Z. Feng. 2014. Two-machine no-wait flowshop scheduling with learning effect and convex resource-dependent processing times. Computers & Industrial Engineering 75: 170-175.

Lu, Y.-Y., G. Li, Y.-B. Wu, and P. Ji. 2014. Optimal due-date assignment problem with learning effect and resource-dependent processing times. Optimization Letters 8 (1): 113-127.

Niu, Y.-P., L. Wan, and J.-B. Wang. 2015. A note on scheduling jobs with extended sum-of-processing-times-based andposition-based learning effect. Asia-Pacific Journal of Operational Research 32(2): 1550001 (12 pages).

Qin, H., Z.-H., Zhang, and D. Bai. 2016. Permutation flowshop group scheduling with position-based learning effect. Computers & Industrial Engineering 92: 1-15.

Rudek, A., and R. Rudek. 2011. A note on optimization in deteriorating systems using scheduling problems with the aging effect and resource allocation models. Computers & Mathematics with Applications 62: 1870-1878.

Rudek, A., and R. Rudek. 2012. On flowshop scheduling problems with the aging effect and resource allocation. International Journal of Advanced Manufacturing Technology 62: 135-145.

Shabtay, D., and G. Steiner. 2007. A survey of scheduling with controllable processing times. Discrete Applied Mathematics 155 (13): 1643-1666.

Shabtay, D., Y. Itskovich, L. Yedidsion, and D. Oron. 2010. Optimal due date assignment and resource allocation in a group technology scheduling environment. Computers & Operations Research 37: 2218-2228.

Shiau, Y.-R., M.-S. Tsai, W.-C. Lee, and T.C.E. Cheng. 2015. Two-agent two-machine flowshop scheduling with learning effects to minimize the total completion time. Computers & Industrial Engineering 87: 580-589.

Wang, X., and T.C.E. Cheng. 2005. Single machine scheduling with resource dependent release times and processing times. European Journal of Operational Research 162: 727C739.

Wang, J.-B., and J.-J. Wang. 2014a. Single machine group scheduling with time dependent processing times and ready times. Information Sciences 275: 226-231.

Wang, J.-B., M.-Z. Wang. 2014b. Single-machine due-window assignment and scheduling with learning effect and resource-dependent processing times. Asia-Pacific Journal of Operational Research 31(5): 1450036 (15 pages).

Wang, J.-B., and J.-J. Wang. 2015. Research on scheduling with job-dependent learning effect and convex resource dependent processing times. International Journal of Production Research 53 (19): 5826-5836.

Wang, D., M.-Z. Wang, J.-B. Wang. 2010. Single-machine scheduling with learning effect and resource-dependent processing times. Computers & Industrial Engineering 59 (3): 458-462.

Xu, J., C.-C. Wu, Y. Yin, C. Zhao, Y.-T. Chiou, and W.-C. Lin. 2016. An order scheduling problem with position-based learning effect. Computers & Operations Research 74: 175-186.

Yang, D-L., C-J. Lai, and S.-J. Yang. 2014. Scheduling problems with multiple due windows assignment and controllable processing times on a single machine. International Journal of Production Economics 150: 96-103.

Yin, N., L. Kang, and X.-Y. Wang. 2014. Single-machine group scheduling with processing times dependent on position, starting time and allotted resource. Applied Mathematical Modelling 38 (19-20): 4602-4613.

Zhang X, and Y. Wang. 2015. Single-machine scheduling CON/SLK due window assignment problems with sum-of-processed times based learning effect. Applied Mathematics and Computation 250 : 628-635.

Zhu, Z., L. Sun, F. Chu, and M. Liu. 2011. Single-machine group scheduling with resource allocation and learning effect. Computers & Industrial Engineering 60 (1): 148-157.