# Scheduling jobs with controllable processing time, truncated job-dependent learning and deterioration effects

Ji-Bo Wang[a], Mengqi Liu[b,1], Na Yin[a], Ping Ji[c]

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

[b]Business School, Hunan University, Changsha 410082, Hunan, China;

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

**Abstract**

In this paper, we consider single machine scheduling problems with controllable processing time (resource allocation), truncated job-dependent learning and deterioration effects. The goal is to find the optimal sequence of jobs and the optimal resource allocation separately for minimizing a cost function containing makespan (total completion time, total absolute differences in completion times) and/or total resource cost. For two different processing time functions, i.e., a linear and a convex function of the amount of a common continuously divisible resource allocated to the job, we solve them in polynomial time respectively.

**Keywords:** Scheduling; Resource allocation; Deterioration job; Learning effect

## 1 Introduction

Scheduling problems jobs with variable job processing times (for example, learning effect and/or deterioration effect and/or resource allocation) have received a lot of attention in the recent academic literatures (Wang et al. (2010), Yang and Kuo (2010), Cheng et al. (2011), Wu et al. (2011), Bai et al. (2012a), Bai et al. (2012b), Wei et al. (2012), Wu et al. (2012), Cheng et al. (2013), Li et al. (2013), Low and Lin (2013), Qian and Steiner (2013), Sun et al. (2013), Wang and Wang (2013a), Wang and Wang (2013b), Wang and Wang (2013c), Wang et al. (2013a), Wang et al. (2013b), Wu et al. (2013), Yang (2013), Guo et al. (2014), Huang et al. (2014), Wang et al. (2014), Wu et al. (2014), Yang et al. (2014a), Yang et al. (2014b), Yin et al. (2014a), Yin et al. (2014b), Li et al. (2015), Niu et al. (2015), and Kacem and Levner (2016)). For more details on scheduling problems and models with learning effects, the reader may refer to the recent survey by Biskup (2008). For more details on scheduling with time-dependent scheduling, the reader may refer to the recent book by Gawiejnowicz (2008). For more details on scheduling with controllable processing times (resource allocation), the reader may refer to the recent paper by Shabtay and Steiner (2007).

[1] Corresponding author. E-mail addresses: wangjibo75@163.com (Ji-Bo Wang), liumengqi163@126.com (Mengqi Liu), shenyinyin@126.com (Na Yin)

Biskup (1999) assumed that the actual processing time of job $J_j$ is $p_{jr}^A = p_j r^a$ where $a \leq 0$ is the learning index, $r$ is a position in a sequence, and $p_j$ is the normal processing time of job $J_j$. Wu et al. (2011) studied a truncated learning effect model, i.e., $p_{jr}^A = p_j \max\{r^a, b\}$, where $b$ is a truncation parameter with $0 < b < 1$. Mosheiov and Sidney (2003) studied job-dependent learning effects, i.e., $p_{jr}^A = p_j r^{a_j}$, where $a_j \leq 0$ is the job-dependent learning index of job $J_j$. Wang et al. [31] studied truncated job-dependent learning effect model, i.e., $p_{jr}^A = p_j \max\{r^{a_j}, b\}$. Yang and Kuo (2010) studied job-dependent learning effect and deteriorating jobs, i.e., $p_{jr}^A = p_j r^{a_j} + ct$, where $t$ is the starting time of job $J_j$ and $c \geq 0$ is a deterioration rate. Niu et al. (2015) studied effects of deterioration and truncated job-dependent learning, i.e., $p_{jr}^A(t) = p_j \max\{r^{a_j}, b\} + ct$. Shabtay and Steiner (2007) considered the linear resource consumption model $p_j^A(u_j) = p_j - \theta_j u_j$, where $u_j$ is the amount resource allocated to job $J_j$ and $\theta_j$ is the positive compression rate of job $J_j$. For a convex resource consumption function: $p_j^A(u_j) = \left(\frac{p_j}{u_j}\right)^l$, $u_j > 0$, where $l$ is a positive constant.

In this paper we combine truncated job-dependent learning effect with deterioration effect and resource allocation, which are either linear or convex resource consumption functions of the amounts of a common continuously divisible resource allocated to the jobs, i.e., we propose a new model stem from Shabtay and Steiner (2007) and Niu et al. (2015). "*The phenomena of deterioration, truncated job-dependent learning effect, and resource allocation occurring simultaneously can be found in many real-life situations. For example, in steel production, more precisely, in the process of pre-heating ingots by gas to prepare them for hot rolling on the blooming mill. Before the ingots can be hot rolled, they have to achieve the required temperature. However, the pre-heating time of the ingots depends on their starting temperature, i.e., the longer ingots wait for the start of the pre-heating process, the lower goes their temperature and therefore the longer lasts the pre-heating process. The pre-heating time can be shortened by the increase of the gas flow intensity, i.e., the more gas is consumed, the shorter lasts the pre-heating process. Thus, the ingot pre-heating time depends on the starting moment of the pre-heating process and the amount of gas consumed during it (Bachman and Janiak (2000)). On the other hand, the truncated job-dependent learning effect reflects that the workers become more skilled to operate the machines through experience accumulation, but the learning effect is controlled will not drop to zero precipitously as the number of jobs increases. For this situation, considering these, the job deterioration, truncated job-dependent learning effect, and resource allocation in job scheduling is both necessary and reasonable* (Wang et al. (2012), and Niu et al. (2015)).

The remaining part of this study is organized as follows. In Section 2, we present the model. In Sections 3 and 4, we consider the scheduling problems for the linear and convex resource consumption functions respectively. The last section presents the conclusions.

## 2 Problems description

The general problem we study in this paper may be stated as follows: $n$ independent and non-preemptive jobs $J = \{J_1, J_2, \ldots, J_n\}$ are immediately available for processing on a single machine. In this paper, we consider two new resource allocation, deterioration effect and truncated job-dependent learning effect models, i.e.,

A linear resource consumption function:

$$p_{jr}^A(t, u_j) = p_j \max\{r^{a_j}, b\} + ct - \theta_j u_j, r, j = 1, 2, \ldots, n, \tag{1}$$

where $p_j$ is a normal processing time of $J_j$, $a_j \leq 0$ is the learning rate of job $J_j$, $b$ is a truncation parameter with $0 < b < 1$, $c$ is a deterioration rate, $t \geq 0$ is the starting time for job $J_j$, $u_j$ is the amount of a non-renewable resource allocated to job $J_j$, with $0 \leq u_j \leq \bar{u}_j < \frac{p_j \max\{n^{a_j}, b\}}{\theta_j}$, where $\bar{u}_j$ denotes the maximum amount of resource allocated to job $J_j$ and $\theta_j$ is the positive compression rate of job $J_j$.

A convex resource consumption function:

$$p_{jr}^A(t, u_j) = \left(\frac{p_j \max\{r^{a_j}, b\}}{u_j}\right)^l + ct, u_j > 0, r, j = 1, 2, \ldots, n. \tag{2}$$

For a given schedule $\pi = [J_1, J_2, \ldots, J_n]$, let $C_j = C_j(\pi)$ be the completion time of job $J_j$. Let $C_{\max} = \max\{C_j | j = 1, 2, \ldots, n\}$ be the makespan; $\sum C_j = \sum_{j=1}^n C_j$ be the total completion time; $TADC = \sum_{i=1}^n \sum_{j=i}^n |C_j - C_i|$ be the total absolute deviation in completion time. Our goal is to determine the optimal resource allocations and the optimal sequence of jobs so that the following cost functions be optimal:

$$Z(\pi, u) = \delta_1 \rho + \delta_2 \sum_{j=1}^n v_j u_j, \tag{3}$$

where weights $\delta_1 \geq 0, \delta_2 \geq 0$ are given constants, $v_j$ is the per time unit cost associated with the resource allocation, $\rho \in \{C_{\max}, \sum C_j, TADC\}$. Using the three-field notation of Graham et al. (1979), the problems can be denoted as $1|B|\delta_1 \rho + \delta_2 \sum_{j=1}^n v_j u_j$, where $B \in \{p_{jr}^A(t, u_j) = p_j \max\{r^{a_j}, b\} + ct - \theta_j u_j, p_{jr}^A(t, u_j) = \left(\frac{p_j \max\{r^{a_j}, b\}}{u_j}\right)^l + ct\}$, and $\rho \in \{C_{\max}, \sum C_j, TADC\}$.

## 3 Linear resource allocation problem

Let $C_{[k]}$ ($p_{[k]}^A$) denote the completion (actual processing) time of the $k$th job in a sequence, by mathematical induction method, we have

**Lemma 1** *For the problem $1|p_{jr}^A(t, u_j) = p_j \max\{r^{a_j}, b\} + ct - \theta_j u_j | \delta_1 \rho + \delta_2 \sum_{j=1}^n v_j u_j$, the completion time and the actual processing time of the $k$th job in a sequence are:*

$$C_{[k]} = \left(p_{[k]} \max\{k^{a_{[k]}}, b\} - \theta_{[k]} u_{[k]}\right) + (1 + c)\left(p_{[k-1]} \max\{(k-1)^{a_{[k-1]}}, b\} - \theta_{[k-1]} u_{[k-1]}\right)$$

$$
\begin{aligned}
& + \ldots + (1+c)^{k-1}\left(p_{[1]} \max\{1^{a_{[1]}}, b\} - \theta_{[1]} u_{[1]}\right) \\
= & \sum_{j=1}^{k} (1+c)^{k-j}\left(p_{[j]} \max\{j^{a_{[j]}}, b\} - \theta_{[j]} u_{[j]}\right); \quad (4)
\end{aligned}
$$

$$
p_{[k]}^{A} = \left(p_{[k]} \max\{k^{a_{[k]}}, b\} - \theta_{[k]} u_{[k]}\right) + c \sum_{j=1}^{k-1} (1+c)^{k-1-j}\left(p_{[j]} \max\{j^{a_{[j]}}, b\} - \theta_{[j]} u_{[j]}\right). \quad (5)
$$

From Eqs. (1), (3), (4) and (5), we have

$$
\begin{aligned}
Z(\pi, u) &= \delta_1 \rho + \delta_2 \sum_{j=1}^{n} v_{[j]} u_{[j]} \\
&= \delta_1 \sum_{j=1}^{n} \omega_j p_{[j]}^{A} + \delta_2 \sum_{j=1}^{n} v_{[j]} u_{[j]} \\
&= \delta_1 \sum_{j=1}^{n} \omega_j \left(\left(p_{[j]} \max\{j^{a_{[j]}}, b\} - \theta_{[j]} u_{[j]}\right) + c \sum_{h=1}^{j-1} (1+c)^{j-1-h}\left(p_{[h]} \max\{h^{a_{[h]}}, b\} - \theta_{[h]} u_{[h]}\right)\right) \\
&\quad + \delta_2 \sum_{j=1}^{n} v_{[j]} u_{[j]} \\
&= \delta_1 \sum_{j=1}^{n} \Omega_j p_{[j]} \max\{j^{a_{[j]}}, b\} + \sum_{j=1}^{n} (\delta_2 v_{[j]} - \theta_{[j]} \Omega_j) u_{[j]}, \quad (6)
\end{aligned}
$$

where $\omega_r = 1$ for $C_{\max}$, $\omega_r = (n-r+1)$ for $\sum C_j$, $\omega_r = (r-1)(n-r+1)$ for $TADC$ (Kanet (1981)) and

$$
\begin{aligned}
\Omega_1 &= \omega_1 + c\omega_2 + c(1+c)\omega_3 + \ldots + c(1+c)^{n-2}\omega_n, \\
\Omega_2 &= \omega_2 + c\omega_3 + c(1+c)\omega_4 + \ldots + c(1+c)^{n-3}\omega_n, \\
\Omega_3 &= \omega_3 + c\omega_4 + c(1+c)\omega_5 \ldots + c(1+c)^{n-4}\omega_n, \quad (7) \\
& \ldots \\
\Omega_{n-1} &= \omega_{n-1} + c\omega_n, \\
\Omega_n &= \omega_n.
\end{aligned}
$$

**Lemma 2** *For a given sequence of the problem* $1|p_{jr}^{A}(t, u_j) = p_j \max\{r^{a_j}, b\} + ct - \theta_j u_j|\delta_1\rho + \delta_2 \sum_{j=1}^{n} v_j u_j$, *where* $\rho \in \{C_{\max}, \sum C_j, TADC\}$, *the optimal resource allocation can be determined as follows:*

$$
u_{[j]}^{*} = \begin{cases} 0, & \text{if } \delta_2 v_{[j]} - \theta_{[j]}\Omega_j > 0, \\ u_{[j]}, & \text{if } \delta_2 v_{i[j]} - \theta_{[j]}\Omega_j = 0, \\ \bar{u}_{[j]}, & \text{if } \delta_2 v_{[j]} - \theta_{[j]}\Omega_j < 0, \end{cases} \quad (8)
$$

*where* $0 \le u_{[j]} \le \bar{u}_{[j]}$ *and* $u_{[j]}^{*}, j = 1, 2, \ldots, n$, *represents the optimal resource allocation of the job in position* $j$.

**Proof.** For the problem $1|p_{jr}^A(t,u_j)=p_j\max\{r^{a_j},b\}+ct-\theta_j u_j|\delta_1\rho+\delta_2\sum_{j=1}^n v_j u_j$, taking the derivative by $u_{[j]}$ to Eq. (6), we have $\frac{df(\pi,u)}{du_{[j]}}=\delta_2 v_{[j]}-\theta_{[j]}\Omega_j$ for $j=1,2,\ldots,n$. Then, for any sequence, the optimal resource allocation of a job in a position with a negative $\delta_2 v_{[j]}-\theta_{[j]}\Omega_j$ should be its upper bound on the amount of resource $\bar{u}_{[j]}$, and the optimal resource allocation of a job in a position with a positive $\delta_2 v_{[j]}-\theta_{[j]}\Omega_j$ should be 0. If $\delta_2 v_{[j]}-\theta_{[j]}\Omega_j=0$, then the optimal resource allocation of the job in this position may be any value between 0 and $\bar{u}_{[j]}$. □

In order to obtain the optimal sequence, we formulate the problem $1|p_{jr}^A(t,u_j)=p_j\max\{r^{a_j},b\}+ct-\theta_j u_j|\delta_1\rho+\delta_2\sum_{j=1}^n v_j u_j$, where $\rho\in\{C_{\max},\sum C_j,TADC\}$, as an assignment problem.

Let

$$\Lambda_{jr}=\begin{cases}\delta_1\Omega_r p_j\max\{r^{a_j},b\}, & \text{if} \quad \delta_2 v_j-\theta_j\Omega_r\geq 0,\\ \delta_1\Omega_r p_j\max\{r^{a_j},b\}+(\delta_2 v_j-\theta_j\Omega_r)\bar{u}_j, & \text{if} \quad \delta_2 v_j-\theta_j\Omega_r<0,\end{cases} \tag{9}$$

and $X_{jr}$ be a binary variable which equals 1 if job $J_j$ is scheduled in position $r$, and 0, otherwise. Hence, the optimal matching of jobs to positions requires a solution for the following assignment problem:

$$\text{Min}\sum_{j=1}^n\sum_{r=1}^n\Lambda_{jr}X_{jr} \tag{10}$$

Subject to

$$\sum_{r=1}^n X_{jr}=1, \quad j=1,2,\ldots,n,$$

$$\sum_{j=1}^n X_{jr}=1, \quad r=1,2,\ldots,n,$$

$$X_{jr}=0 \text{ or } 1, \quad j,r=1,2,\ldots,n.$$

Recall that solving an assignment problem requires $O(n^3)$ time (using the well-known Hungarian method).

Based on the above analysis, we can obtain that the problem $1|p_{jr}^A(t,u_j)=p_j\max\{r^{a_j},b\}+ct-\theta_j u_j|\delta_1\rho+\delta_2\sum_{j=1}^n v_j u_j$ ($\rho\in\{C_{\max},\sum C_j,TADC\}$) can be solved by the following algorithm:

**Algorithm 1**

*Step 1.* Calculate $\Lambda_{jr}$ for $r,j=1,2\ldots,n$ by using Eq. (9).

*Step 2.* Solve the assignment problem (10) to determine the optimal job sequence.

*Step 3.* Calculate the optimal resource allocations by using Eq. (8).

*Step 4.* Calculate the optimal processing times by using Eq. (1).

**Theorem 1** *Algorithm 1 solves the problem* $1|p_{jr}^A(t,u_j)=p_j\max\{r^{a_j},b\}+ct-\theta_j u_j|\delta_1\rho+\delta_2\sum_{j=1}^n v_j u_j$ *in* $O(n^3)$ *time, where* $\rho\in\{C_{\max},\sum C_j,TADC\}$.

We present the following example to illustrate Algorithm 1 for the problem $1|p_j^A(t,u_j) = p_j \max\{r^{a_j}, b\} + ct - \beta_j u_j|\delta_1 \sum_{j=1}^n C_j + \delta_2 \sum_{j=1}^n v_j u_j$.

**Example 1** There are $n = 6$ jobs to be processed on a single machine, where $b = 0.7, l = 2, c = 0.05, \delta_1 = \delta_2 = 1$, and other corresponding data are shown in Table 1. From Eq. (7), we have $\Omega_1 = 6.8019$, $\Omega_2 = 5.5256$, $\Omega_3 = 4.3101$, $\Omega_4 = 3.1525$, $\Omega_5 = 2.0500$, $\Omega_6 = 1$.

**Solution**

*Step 1.* From Eq. (9), the values $\Lambda_{jr}, r, j = 1, 2 \ldots, n$ are given in Table 2.

*Step 2.* Solve the assignment problem (11), we obtain that the optimal job sequence is $\pi^* = [J_3, J_5, J_6, J_2, J_1, J_4]$.

*Step 3.* From Eq. (8), the optimal resource allocations are $u_1 = 0, u_2 = 0, u_3 = 3, u_4 = 0, u_5 = 2, u_6 = 2$.

*Step 4.* From Eq. (1), the optimal processing times are $p_1^A = 7.7312, p_2^A = 6.8849, p_3^A = 2.0000, p_4^A = 16.1650, p_5^A = 1.4103, p_6^A = 4.3279$, and the total cost is $\delta_1 \sum_{j=1}^n C_j + \delta_2 \sum_{j=1}^n v_j u_j$=170.6448.

Table 1 Data of Example 1

| $J_j$ | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ | $J_6$ |
|---|---|---|---|---|---|---|
| $p_j$ | 10 | 8 | 11 | 18 | 9 | 16 |
| $\beta_j$ | 2 | 1 | 3 | 2 | 3 | 4 |
| $\bar{u}_j$ | 3 | 2 | 3 | 1 | 2 | 2 |
| $v_j$ | 10 | 8 | 12 | 11 | 14 | 9 |
| $a_j$ | -0.25 | -0.15 | -0.2 | -0.1 | -0.3 | -0.25 |

Table 2 Values of $\Lambda_{jr}$

| $j\backslash r$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 57.2076 | 43.3110 | 32.7497 | 22.2915 | **14.3500** | 7.0000 |
| 2 | 54.4152 | 39.8396 | 29.2421 | **20.4850** | 12.8824 | 6.1146 |
| 3 | **49.6038** | 39.1831 | 35.2680 | 26.2806 | 16.3438 | 7.7000 |
| 4 | 119.8304 | 92.7490 | 69.5101 | 49.3994 | 31.4144 | **15.0473** |
| 5 | 48.4057 | **35.2400** | 27.8993 | 19.8607 | 12.9150 | 6.3000 |
| 6 | 72.4152 | 48.1385 | **35.9187** | 28.4465 | 22.9600 | 11.2000 |

The bold numbers are the optimal solution

## 4 Convex resource allocation problem

### 4.1 Problem $1|p_{jr}^A(t,u_j)=\left(\frac{p_j\max\{r^{a_j},b\}}{u_j}\right)^l+ct|\delta_1\rho+\delta_2\sum_{j=1}^n v_ju_j$

**Lemma 3** *For the problem* $1|p_{jr}^A(t,u_j)=\left(\frac{p_j\max\{r^{a_j},b\}}{u_j}\right)^l+ct|\delta_1\rho+\delta_2\sum_{j=1}^n v_ju_j$, *the completion time and the actual processing time of the kth job in a sequence are:*

$$\begin{aligned}
C_{[k]} &= \left(\frac{p_{[k]}\max\{k^{a_{[k]}},b\}}{u_{[k]}}\right)^l+(1+c)\left(\frac{p_{[k-1]}\max\{(k-1)^{a_{[k-1]}},b\}}{u_{[k-1]}}\right)^l\\
&\quad +\ldots+(1+c)^{k-1}\left(\frac{p_{[1]}\max\{1^{a_{[1]}},b\}}{u_{[1]}}\right)^l\\
&= \sum_{j=1}^{k}(1+c)^{k-j}\left(\frac{p_{[k]}\max\{k^{a_{[k]}},b\}}{u_{[k]}}\right)^l; \qquad (11)\\
p_{[k]}^A &= \left(\frac{p_{[k]}\max\{k^{a_{[k]}},b\}}{u_{[k]}}\right)^l+c\sum_{j=1}^{k-1}(1+c)^{k-1-j}\left(\frac{p_{[j]}\max\{j^{a_{[j]}},b\}}{u_{[j]}}\right)^l. \qquad (12)
\end{aligned}$$

From Eqs. (2), (3), (11) and (12), we have

$$\begin{aligned}
Z(\pi,u) &= \delta_1\rho+\delta_2\sum_{j=1}^n v_{[j]}u_{[j]}\\
&= \delta_1\sum_{j=1}^n\omega_jp_{[j]}^A+\delta_2\sum_{j=1}^n v_{[j]}u_{[j]}\\
&= \delta_1\sum_{j=1}^n\omega_j\left(\left(\frac{p_{[j]}\max\{j^{a_{[j]}},b\}}{u_{[j]}}\right)^l+c\sum_{h=1}^{j-1}(1+c)^{j-1-h}\left(\frac{p_{[h]}\max\{h^{a_{[h]}},b\}}{u_{[h]}}\right)^l\right)\\
&\quad +\delta_2\sum_{j=1}^n v_{[j]}u_{[j]}\\
&= \delta_1\sum_{j=1}^n\Omega_j\left(\frac{p_{[j]}\max\{j^{a_{[j]}},b\}}{u_{[j]}}\right)^l+\delta_2\sum_{j=1}^n v_{[j]}u_{[j]}, \qquad (13)
\end{aligned}$$

where $\Omega_j$ are given by Eq. (7).

In the following lemma, we determine the optimal resource allocation $u^*(\pi)$.

**Lemma 4** *For the problem* $1|p_{jr}^A(t,u_j)=\left(\frac{p_j\max\{r^{a_j},b\}}{u_j}\right)^l+ct|\delta_1\rho+\delta_2\sum_{j=1}^n v_ju_j$, *where* $\rho\in\{C_{\max},\sum C_j,TADC\}$, *the optimal resource allocation as a function of the job sequence, i.e.,*

$$u_{[j]}^*=\left(\frac{l\delta_1\Omega_j}{\delta_2v_{[j]}}\right)^{\frac{1}{l+1}}\times\left(p_{[j]}\max\{j^{a_{[j]}},b\}\right)^{\frac{l}{l+1}}, \qquad (14)$$

*where* $\Omega_j$ *are given by Eq. (7).*

**Proof.** By taking the derivative given by Eq. (13) with respect to $u_{[j]}, j = 1, 2, \ldots, n$, equating it to zero and solving it for $u_{[j]}$, we obtain Eq. (14). □

By substituting Eq. (14) into Eq. (13), we have

$$Z(\pi, u^*(\pi)) = \left(l^{\frac{-l}{l+1}} + l^{\frac{1}{l+1}}\right) \delta_2^{\frac{l}{l+1}} \times \sum_{j=1}^{n} \left(v_{[j]}p_{[j]}\right)^{\frac{l}{l+1}} \left(\delta_1 \Omega_j\right)^{\frac{1}{l+1}} \left(\max\left\{j^{a_{[j]}}, b\right\}\right)^{\frac{l}{l+1}}, \tag{15}$$

where $\Omega_j$ are given by Eq. (7).

**Lemma 5** *The optimal job sequence for the problem* $1|p_{jr}^{A}(t, u_j) = \left(\frac{p_j \max\{r^{a_j}, b\}}{u_j}\right)^l + ct|\delta_1 \rho + \delta_2 \sum_{j=1}^{n} v_j u_j$ *(*$\rho \in \{C_{\max}, \sum C_j, TADC\}$*) can be determined in* $O(n^3)$ *time by the following linear assignment problem:*

$$\text{Min} \sum_{j=1}^{n} \sum_{r=1}^{n} \Theta_{jr} X_{jr} \tag{16}$$

Subject to

$$\sum_{r=1}^{n} X_{jr} = 1, \quad j = 1, 2, \ldots, n,$$

$$\sum_{j=1}^{n} X_{jr} = 1, \quad r = 1, 2, \ldots, n,$$

$$X_{jr} = 0 \text{ or } 1, \quad j, r = 1, 2, \ldots, n.$$

*where*

$$\Theta_{jr} = \left(l^{\frac{-l}{l+1}} + l^{\frac{1}{l+1}}\right) \delta_2^{\frac{l}{l+1}} \left(v_j p_j\right)^{\frac{l}{l+1}} \left(\delta_1 \Omega_r\right)^{\frac{1}{l+1}} \left(\max\left\{r^{a_j}, b\right\}\right)^{\frac{l}{l+1}}. \tag{17}$$

Hence the problem $1|p_{jr}^{A}(t, u_j) = \left(\frac{p_j \max\{r^{a_j}, b\}}{u_j}\right)^l + ct|\delta_1 \rho + \delta_2 \sum_{j=1}^{n} v_j u_j$ ($\rho \in \{C_{\max}, \sum C_j, TADC\}$) can be solved by the following optimization algorithm.

**Algorithm 2**

*Step 1.* Calculate $\Theta_{jr}$ for $j, r = 1, 2, \ldots, n$ by Eq. (17).

*Step 2.* Sequence the jobs according to Lemma 5, and denote the resulting optimal sequence by $\pi^* = [J_{[1]}, J_{[2]}, \ldots, J_{[n]}]$.

*Step 3.* Calculate the optimal resources by using Eq. (14).

*Step 4.* Calculate the optimal processing times by using Eq. (2).

**Theorem 2** *Algorithm 2 solves the scheduling problem* $1|p_{jr}^{A}(t, u_j) = \left(\frac{p_j \max\{r^{a_j}, b\}}{u_j}\right)^l + ct|\delta_1 \rho + \delta_2 \sum_{j=1}^{n} v_j u_j$ *(*$\rho \in \{C_{\max}, \sum C_j, TADC\}$*) in* $O(n^3)$ *time.*

We present the following example to illustrate Algorithm 1 for the problem $1|p_j^A(t,u_j) = \left(\frac{p_j \max\{r^{a_j},b\}}{u_j}\right)^l + ct|\delta_1 \sum_{j=1}^n C_j + \delta_2 \sum_{j=1}^n v_j u_j$.

**Example 2** There are $n = 6$ jobs to be processed on a single machine, where $l = 2, b = 0.7, c = 0.05, \delta_1 = \delta_2 = 1$, and other corresponding data are shown in Table 3. From Example 1, we have $\Omega_1 = 6.8019$, $\Omega_2 = 5.5256$, $\Omega_3 = 4.3101$, $\Omega_4 = 3.1525$, $\Omega_5 = 2.0500$, $\Omega_6 = 1$.

**Solution**

*Step 1.* From Eq. (17), the values $\Theta_{jr}, r, j = 1, 2 \ldots, n$ are given in Table 4.

*Step 2.* Solve the assignment problem (16), we obtain that the optimal job sequence is $\pi^* = [J_2, J_1, J_5, J_6, J_3, J_4]$.

*Step 3.* From Eq. (14), the optimal resource allocations are $u_2 = \left(\frac{l\delta_1\Omega_j}{\delta_2 v_{[j]}}\right)^{\frac{1}{l+1}} \times \left(p_{[j]} \max\left\{j^{a_{[j]}}, b\right\}\right)^{\frac{l}{l+1}} = \left(\frac{2*6.8019}{8}\right)^{\frac{1}{3}} \times \left(8 \max\left\{1^{-0.15}, 0.7\right\}\right)^{\frac{2}{3}} = 4.7744, u_1 = 4.2753, u_5 = 2.9548, u_6 = 4.4759, u_3 = 2.7900, u_4 = 4.3863$.

*Step 4.* From Eq. (2), the optimal processing times are $p_2^A = 2.8076, p_1^A = 4.0090, p_5^A = 5.1399, p_6^A = 6.9871, p_3^A = 9.1128, p_4^A = 13.1712$, and the total cost is $\delta_1 \sum_{j=1}^n C_j + \delta_2 \sum_{j=1}^n v_j u_j$=351.0909.

Table 3. Data of Example 2

| $J_j$ | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ | $J_6$ |
|---|---|---|---|---|---|---|
| $p_j$ | 10 | 8 | 11 | 18 | 9 | 16 |
| $v_j$ | 10 | 8 | 12 | 11 | 14 | 9 |
| $a_j$ | -0.25 | -0.15 | -0.2 | -0.1 | -0.3 | -0.25 |

Table 4. Values of $\Theta_{jr}$

| $j \backslash r$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 77.1456 | **64.1292** | 55.1751 | 47.3852 | 40.7772 | 32.0996 |
| 2 | **57.2925** | 49.8783 | 44.0898 | 38.5982 | 32.7021 | 25.2778 |
| 3 | 92.8312 | 78.9720 | 68.8700 | 59.7165 | **50.2195** | 38.6263 |
| 4 | 121.6433 | 108.3767 | 97.1029 | 85.8274 | 73.2596 | **56.9729** |
| 5 | 89.9964 | 73.1030 | **62.0516** | 54.9075 | 47.5698 | 37.4467 |
| 6 | 98.3754 | 81.7770 | 70.3588 | **60.4252** | 51.9987 | 40.9331 |

The bold numbers are the optimal solution

In the following, we present a simpler and more efficient solution for a special case, i.e., $a_j = a$ for $j = 1, 2, \ldots, n$. Stem from (16), we have

$$Z(\pi, u^*(\pi)) = \left(l^{\frac{-l}{l+1}} + l^{\frac{1}{l+1}}\right) (\delta_1)^{\frac{1}{l+1}} (\delta_2)^{\frac{l}{l+1}} \times \sum_{j=1}^{n} \left(v_{[j]}p_{[j]}\right)^{\frac{l}{l+1}} (\Omega_j \max\{j^a, b\})^{\frac{1}{l+1}}, \tag{18}$$

where $\Omega_j$ are given by Eq. (7).

In order to find the job sequence that minimizes $Z(\pi, u^*(\pi))$, we have to optimally match $(\Omega_j \max\{j^a, b\})^{\frac{1}{l+1}}$ with $\left(v_{[j]}p_{[j]}\right)^{\frac{l}{l+1}}$. It is well known that $\sum_{j=1}^{n} \left(v_{[j]}p_{[j]}\right)^{\frac{l}{l+1}} (\Omega_j \max\{j^a, b\})^{\frac{1}{l+1}}$ is minimized by sorting the elements of the $(\Omega_j \max\{j^a, b\})^{\frac{1}{l+1}}$ and $\left(v_{[j]}p_{[j]}\right)^{\frac{l}{l+1}}$ vectors in opposite orders (i.e., HLP principle, Hardy et al. (1976)). Hence, we have

**Lemma 6** *The optimal job sequence for the problem $1|p_{jr}^A(t, u_j) = \left(\frac{p_j \max\{r^{a_j}, b\}}{u_j}\right)^l + ct, a_j = a|\delta_1\rho + \delta_2 \sum_{j=1}^{n} v_j u_j$ ($\rho \in \{C_{\max}, \sum C_j, TADC\}$) can be determined in $O(n \log n)$ time by the HLP principle (Hardy et al. (1976)).*

The results of our analysis are summarized in the following optimization algorithm that solves the problem $1|p_{jr}^A(t, u_j) = \left(\frac{p_j \max\{r^{a_j}, b\}}{u_j}\right)^l + ct, a_j = a|\delta_1\rho + \delta_2 \sum_{j=1}^{n} v_j u_j$ ($\rho \in \{C_{\max}, \sum C_j, TADC\}$).

**Algorithm 3**

*Step 1.* Calculate $(\Omega_j \max\{j^a, b\})^{\frac{1}{l+1}}$ and $(v_j p_j)^{\frac{l}{l+1}}$ for $j, r = 1, 2, \ldots, n$.

*Step 2.* Sequence the jobs by the HLP principle (Lemma 6), and denote the resulting optimal sequence by $\pi^* = [J_{[1]}, J_{[2]}, \ldots, J_{[n]}]$.

*Step 3.* Calculate the optimal resources by using Eq. (14).

*Step 4.* Calculate the optimal processing times by using Eq. (2).

**Theorem 3** *Algorithm 3 solves the scheduling problem $1|p_{jr}^A(t, u_j) = \left(\frac{p_j \max\{r^{a_j}, b\}}{u_j}\right)^l + ct|\delta_1\rho + \delta_2 \sum_{j=1}^{n} v_j u_j$ ($\rho \in \{C_{\max}, \sum C_j, TADC\}$) in $O(n \log n)$ time.*

### 4.2 Problem $1|p_{jr}^A(t, u_j) = \left(\frac{p_j \max\{r^{a_j}, b\}}{u_j}\right)^l + ct, \sum_{j=1}^{n} u_j \le U|\rho$

In this subsection, we consider the problem $1|p_{jr}^A(t, u_j) = \left(\frac{p_j \max\{r^{a_j}, b\}}{u_j}\right)^l + ct, \sum_{j=1}^{n} u_j \le U|\rho$, subject to $\sum_{j=1}^{n} u_j \le U$, where $U > 0$ is a given constant, and $\rho \in \{C_{\max}, \sum C_j, TADC\}$.

**Lemma 7** *For the $1|p_{jr}^A(t, u_j) = \left(\frac{p_j \max\{r^{a_j}, b\}}{u_j}\right)^l + ct, \sum_{j=1}^{n} u_j \le U|\rho$, where $\rho \in \{C_{\max}, \sum C_j, TADC\}$,*

$$u_{[j]}^* = \frac{(\Omega_j)^{1/(l+1)} \left(p_{[j]} \max\{j^{a_{[j]}}, b\}\right)^{l/(l+1)}}{\sum_{j=1}^{n} (\Omega_j)^{1/(l+1)} \left(p_{[j]} \max\{j^{a_{[j]}}, b\}\right)^{l/(l+1)}} \times U \quad (j = 1, 2, \ldots, n), \tag{19}$$

*where $\Omega_j$ are given by Eq. (7).*

**Proof** For a given sequence, the Lagrange function is

$$\begin{aligned} L(\mathbf{u},\lambda) &= \rho+\lambda\left(\sum_{j=1}^{n}u_{[j]}-U\right) \\ &= \sum_{j=1}^{n}\Omega_j\left(\frac{p_{[j]}\max\left\{j^{a_{[j]}},b\right\}}{u_{[j]}}\right)^l+\lambda\left(\sum_{j=1}^{n}u_{[j]}-U\right), \end{aligned} \tag{20}$$

where $\lambda$ is the Lagrangian multiplier. Deriving (20) with respect to $u_{[j]}$ and $\lambda$, we have

$$\frac{\partial L(\mathbf{u},\lambda)}{\partial\lambda}=\sum_{j=1}^{n}u_{[j]}-U=0, \tag{21}$$

$$\frac{\partial L(\mathbf{u},\lambda)}{\partial u_{[j]}}=\lambda-l\Omega_j\times\frac{\left(p_{[j]}\max\left\{j^{a_{[j]}},b\right\}\right)^l}{(u_{[j]})^{l+1}}=0,\ \forall j=1,2,\ldots,n. \tag{22}$$

Using (21) and (22) we obtain

$$u_{[j]}=\frac{\left(k\Omega_j\left(p_{[j]}\max\left\{j^{a_{[j]}},b\right\}\right)^l\right)^{1/(l+1)}}{\lambda^{1/(l+1)}}, \tag{23}$$

and

$$\lambda^{1/(l+1)}=\frac{\sum_{j=1}^{n}(l\Omega_j)^{1/(l+1)}\left(p_{[j]}\max\left\{j^{a_{[j]}},b\right\}\right)^{l/(l+1)}}{U}. \tag{24}$$

From (7) and (8), we have

$$u^*_{[j]}=\frac{(\Omega_j)^{1/(k+1)}\left(p_{[j]}\max\left\{j^{a_{[j]}},b\right\}\right)^{l/(l+1)}}{\sum_{j=1}^{n}(\Omega_j)^{1/(k+1)}\left(p_{[j]}\max\left\{j^{a_{[j]}},b\right\}\right)^{l/(l+1)}}\times U.$$

□

Substituting (19) into $\rho=\sum_{j=1}^{n}\Omega_j\left(\frac{p_{[j]}\max\left\{j^{a_{[j]}},b\right\}}{u_{[j]}}\right)^l$, we have

$$\begin{aligned} \rho(\pi,u^*) &= U^{-l}\left(\sum_{j=1}^{n}(\Omega_j)^{1/(l+1)}\left(p_{[j]}\max\left\{j^{a_{[j]}},b\right\}\right)^{l/(l+1)}\right)^{l+1} \\ &= U^{-l}\left(\sum_{j=1}^{n}\Omega_j\left(p_{[j]}\max\left\{j^{a_{[j]}},b\right\}\right)^l\right), \end{aligned} \tag{25}$$

where $\Omega_j$ are given by Eq. (7).

Similar to Subsection 4.1, we have

**Lemma 8** *The optimal job sequence for the problem* $1|p^A_{jr}(t,u_j)=\left(\frac{p_j\max\left\{r^{a_j},b\right\}}{u_j}\right)^l+ct,\sum_{j=1}^{n}u_j\le U|\rho$ *(*$\rho\in\{C_{\max},\sum C_j,TADC\}$*) can be determined in* $O(n^3)$ *time by the following linear assignment problem:*

$$\text{Min}\sum_{j=1}^{n}\sum_{r=1}^{n}\vartheta_{jr}X_{jr}, \tag{26}$$

Subject to

$$\sum_{r=1}^{n} X_{jr} = 1, \quad j = 1, 2, \ldots, n,$$

$$\sum_{j=1}^{n} X_{jr} = 1, \quad r = 1, 2, \ldots, n,$$

$$X_{jr} = 0 \text{ or } 1, \quad j, r = 1, 2, \ldots, n,$$

*where*

$$\vartheta_{jr} = U^{-l}\Omega_r \left(p_j \max\left\{r^{a_j}, b\right\}\right)^l . \tag{27}$$

The results of our analysis are summarized in the following optimization algorithm that solves the problem $1|p_j^A = \left(\frac{p_j \max\left\{r^{a_j},b\right\}}{u_j}\right)^l + ct, \sum_{j=1}^n u_j \le U|\rho$ ($\rho \in \{C_{\max}, \sum C_j, TADC\}$).

**Algorithm 4**

*Step 1.* Calculate $\vartheta_{jr}$ for $j, r = 1, 2, \ldots, n$ by Eq. (27).

*Step 2.* Sequence the jobs according to Lemma 8, and denote the resulting optimal sequence by $\pi^* = [J_{[1]}, J_{[2]}, \ldots, J_{[n]}]$.

*Step 3.* Calculate the optimal resources by using Eq. (19).

*Step 4.* Calculate the optimal processing times by using Eq. (2).

**Theorem 4** *Algorithm 4 solves the scheduling problem* $1|p_j^A = \left(\frac{p_j \max\left\{r^{a_j},b\right\}}{u_j}\right)^l + ct, \sum_{j=1}^n u_j \le U|\rho$ *($\rho \in \{C_{\max}, \sum C_j, TADC\}$) in $O(n^3)$ time.*

Similar to Subsection 4.1, if $a_j = a$, $j = 1, 2, \ldots, n$, we have

$$\rho(\pi, u^*(\pi)) = U^{-l}\left(\sum_{j=1}^{n} \Omega_j \left(\max\left\{j^a, b\right\}\right)^l \left(p_{[j]}\right)^l\right). \tag{28}$$

where $\Omega_j$ are given by Eq. (7).

**Theorem 5** *The problem* $1|p_j^A = \left(\frac{p_j \max\{r^a,b\}}{u_j}\right)^l + ct, \sum_{j=1}^n u_j \le U|\rho$ *($\rho \in \{C_{\max}, \sum C_j, TADC\}$) can be solved in $O(n \log n)$ time.*

### 4.3 Problem $1|p_{jr}^A(t, u_j) = \left(\frac{p_j \max\left\{r^{a_j},b\right\}}{u_j}\right)^l + ct, \rho \le R|\sum_{j=1}^n u_j$

In this Subsection, we consider the "inverse version" of the problem $1|p_j^A = \left(\frac{p_j \max\left\{r^{a_j},b\right\}}{u_j}\right)^l + ct, \sum_{j=1}^n u_j \le U|\rho$, i.e., we consider the problem $1|p_{jr}^A(t, u_j) = \left(\frac{p_j \max\left\{r^{a_j},b\right\}}{u_j}\right)^l + ct, \rho \le R|\sum_{j=1}^n u_j$, where $R > 0$ is a given constant, and $\rho \in \{C_{\max}, \sum C_j, TADC\}$. Similar to Subsection 4.2, we have

**Lemma 9** *For the problem* $1|p^A_{jr}(t,u_j) = \left(\frac{p_j \max\{r^{a_j},b\}}{u_j}\right)^l + ct, \rho \le R|\sum_{j=1}^n u_j$, *where* $\rho \in \{C_{\max}, \sum C_j, TADC\}$,

$$u^*_{[j]} = (\Omega_j)^{\frac{1}{l+1}} \left(p_{[j]} \max\{j^{a_{[j]}}, b\}\right)^{\frac{l}{l+1}} \left(\sum_{j=1}^n (\Omega_j)^{\frac{1}{l+1}} \left(p_{[j]} \max\{j^{a_{[j]}}, b\}\right)^{\frac{l}{l+1}}\right)^{\frac{1}{l}} \times R^{-\frac{1}{l}}, \qquad (29)$$

*where* $\Omega_j$ *are given by Eq. (7).*

Substituting (29) into $\sum_{j=1}^n u_j$, we have

$$\sum_{j=1}^n u_j(\pi, u^*) = \left(\sum_{j=1}^n (\Omega_j)^{\frac{1}{l+1}} \left(p_{[j]} \max\{j^{a_{[j]}}, b\}\right)^{\frac{l}{l+1}}\right)^{\frac{1}{l}+1} \times R^{-\frac{1}{l}}, \qquad (30)$$

where $\Omega_j$ are given by Eq. (7).

**Lemma 10** *The optimal job sequence for the problem* $1|p^A_{jr}(t,u_j) = \left(\frac{p_j \max\{r^{a_j},b\}}{u_j}\right)^l + ct, \rho \le R|\sum_{j=1}^n u_j$ *(*$\rho \in \{C_{\max}, \sum C_j, TADC\}$*) can be determined in* $O(n^3)$ *time by the following linear assignment problem:*

$$\text{Min} \sum_{j=1}^n \sum_{r=1}^n \eta_{jr} X_{jr} \qquad (31)$$

Subject to

$$\sum_{r=1}^n X_{jr} = 1, \quad j = 1, 2, \ldots, n,$$

$$\sum_{j=1}^n X_{jr} = 1, \quad r = 1, 2, \ldots, n,$$

$$X_{jr} = 0 \text{ or } 1, \quad j, r = 1, 2, \ldots, n.$$

*where*

$$\eta_{jr} = (\Omega_r)^{\frac{1}{k+1}} \left(p_j \max\{r^{a_j}, b\}\right)^{\frac{k}{k+1}}. \qquad (32)$$

**Theorem 6** *The problem* $1|p^A_j = \left(\frac{p_j \max\{r^{a_j},b\}}{u_j}\right)^l + ct, \rho \le R|\sum_{j=1}^n u_j (\rho \in \{C_{\max}, \sum C_j, TADC\})$ *can be solved in* $O(n^3)$ *time.*

**Theorem 7** *The problem* $1|p^A_j = \left(\frac{p_j \max\{r^a,b\}}{u_j}\right)^l + ct, \rho \le R|\sum_{j=1}^n u_j (\rho \in \{C_{\max}, \sum C_j, TADC\})$ *can be solved in* $O(n \log n)$ *time.*

### 4.4 Problem $1|p_j^A = \left(\frac{p_j \max\{r^a, b\}}{u_j}\right)^l + ct|(\rho, \sum_{j=1}^n u_j)$

In this section, we consider the problem of identifying the set of Pareto-optimal solutions for $(\rho, \sum_{j=1}^n u_j)$, where a schedule $\pi$ with $\rho = R$ and $\sum_{j=1}^n u_j = U$ is called Pareto-optimal if there does not exist another schedule $\pi'$ such that $\rho(\pi') \le R$ and $\sum_{j=1}^n u_j(\pi') \le U$ with at least one of these inequalities being strict. Obviously, the Pareto-optimal of the problem $1|p_j^A = \left(\frac{p_j \max\{r^a, b\}}{u_j}\right)^l + ct|(\rho, \sum_{j=1}^n u_j)$ can be obtained by solving the problem $1|p_j^A = \left(\frac{p_j \max\{r^{a_j}, b\}}{u_j}\right)^l + ct, \sum_{j=1}^n u_j \le U|\rho$ for various values of $U$, or the problem $1|p_j^A = \left(\frac{p_j \max\{r^{a_j}, b\}}{u_j}\right)^l + ct, \rho \le R| \sum_{j=1}^n u_j$ for various values of $R$ (see Hoogeveen (2005), and Shabtay and Steiner (2007)). From Eq. (25), the efficient frontier is $\rho(\pi^*, u^*) = U^{-l}\left(\sum_{j=1}^n \Omega_j \left(p_{[j]} \max\{j^{a_{[j]}}, b\}\right)^l\right)$ (a function of $U = \sum_{j=1}^n u_j$ and can be obtained by Algorithm 4), or from Eq. (30), the efficient frontier is $\sum_{j=1}^n u_j(\pi^*, u^*) = \left(\sum_{j=1}^n (\Omega_j)^{\frac{1}{l+1}} \left(p_{[j]} \max\{j^{a_{[j]}}, b\}\right)^{\frac{l}{l+1}}\right)^{\frac{1}{l}+1} \times R^{-\frac{1}{l}}$ (a function of $R = \rho$), hence, we have

**Theorem 8** *The problem $1|p_j^A = \left(\frac{p_j \max\{r^{a_j}, b\}}{u_j}\right)^l + ct|(\rho, \sum_{j=1}^n u_j)$ ($\rho \in \{C_{\max}, \sum C_j, TADC\}$) can be solved in $O(n^3)$ time.*

**Theorem 9** *The problem $1|p_j^A = \left(\frac{p_j \max\{r^a, b\}}{u_j}\right)^l + ct|(\rho, \sum_{j=1}^n u_j)$ ($\rho \in \{C_{\max}, \sum C_j, TADC\}$) can be solved in $O(n \log n)$ time.*

## 5 Conclusions

This article combines truncated job-dependent learning effect with deterioration effect and resource allocation. For linear and convex resource consumption functions, we showed that several single machine scheduling problems can be solved in polynomial time respectively (see Table 5). Future research may focus on considering other regular and non-regular objective functions, studying the flexible flow shop scheduling problems, and investigating scheduling problems with maintenance activities (Xu et al. (2008), and Xu et al. (2015)).

Table 5. Main results of this paper ($\rho \in \{C_{\max}, \sum C_j, TADC\}$)

| Problem | Complexity | Ref. |
|---|---|---|
| $1\vert p_{jr}^{A}(t,u_j)=p_j\max\{r^{a_j},b\}+ct-\theta_j u_j\vert\delta_1\rho+\delta_2\sum_{j=1}^{n}v_j u_j$ | $O(n^3)$ | Theorem 1 |
| $1\vert p_{jr}^{A}(t,u_j)=\left(\frac{p_j\max\left\{r^{a_j},b\right\}}{u_j}\right)^l+ct\vert\delta_1\rho+\delta_2\sum_{j=1}^{n}v_j u_j$ | $O(n^3)$ | Theorem 2 |
| $1\vert p_{jr}^{A}(t,u_j)=\left(\frac{p_j\max\{r^{a},b\}}{u_j}\right)^l+ct\vert\delta_1\rho+\delta_2\sum_{j=1}^{n}v_j u_j$ | $O(n\log n)$ | Theorem 3 |
| $1\vert p_{jr}^{A}(t,u_j)=\left(\frac{p_j\max\left\{r^{a_j},b\right\}}{u_j}\right)^l+ct,\sum_{j=1}^{n}u_j\le U\vert\rho$ | $O(n^3)$ | Theorem 4 |
| $1\vert p_{jr}^{A}(t,u_j)=\left(\frac{p_j\max\{r^{a},b\}}{u_j}\right)^l+ct,\sum_{j=1}^{n}u_j\le U\vert\rho$ | $O(n\log n)$ | Theorem 5 |
| $1\vert p_{jr}^{A}(t,u_j)=\left(\frac{p_j\max\left\{r^{a_j},b\right\}}{u_j}\right)^l+ct,\rho\le R\vert\sum_{j=1}^{n}u_j$ | $O(n^3)$ | Theorem 6 |
| $1\vert p_{jr}^{A}(t,u_j)=\left(\frac{p_j\max\{r^{a},b\}}{u_j}\right)^l+ct,\rho\le R\vert\sum_{j=1}^{n}u_j$ | $O(n\log n)$ | Theorem 7 |
| $1\vert p_{j}^{A}=\left(\frac{p_j\max\left\{r^{a_j},b\right\}}{u_j}\right)^l+ct\vert(\rho,\sum_{j=1}^{n}u_j)$ | $O(n^3)$ | Theorem 8 |
| $1\vert p_{j}^{A}=\left(\frac{p_j\max\{r^{a},b\}}{u_j}\right)^l+ct\vert(\rho,\sum_{j=1}^{n}u_j)$ | $O(n\log n)$ | Theorem 9 |

## 6 Acknowledgements

We wish to thank two anonymous referees for their helpful comments on the earlier version of the article. The work described in this paper was partially supported by the grant from The Hong Kong Polytechnic University (Project G-YBFE) and the National Natural Science Foundation of China (Grant Nos. 71471120 and 71471057).

## References

[1] Bachman, A., and Janiak, A. (2000). Scheduling deteriorating jobs dependent on resources for the makespan minimization. In Operations Research Proceedings 2000: Selected Papers of the Symposium on Operations Research (OR 2000), pp. 29-34 Dresden: Springer.

[2] Bai, J., Li, Z.-R., and Huang, X. (2012a). Single-machine group scheduling with general deterioration and learning effects. Applied Mathematical Modelling, 36, 1267-1274.

[3] Bai, J., Wang, M.-Z., and Wang, J.-B. (2012b). Single machine scheduling with a general exponential learning effect. Applied Mathematical Modelling, 36, 829-835.

[4] Biskup, D. (1999). Single-machine scheduling with learning considerations. European Journal of Operational Research, 115, 173-178.

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

[6] Cheng, T.C.E., Cheng, S.-R., Wu, W.-H., Hsu, P.-H., and Wu, C.-C. (2011). A two-agent single-machine scheduling problem with truncated sum-of-processing-times-based learning considerations. Computers & Industrial Engineering, 60, 534-541.

[7] Cheng, T.C.E., Kuo, W.-H., abd Yang, D.-L. (2013). Scheduling with a position-weighted learning effect based on sum-of-logarithm-processing-times and job position. Information Sciences, 221, 490-500.

[8] Gawiejnowicz, S. (2008). Time-dependent scheduling. Springer, Berlin.

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

[10] Guo, P., Cheng W., and Wang, Y. (2014). A general variable neighborhood search for single-machine total tardiness scheduling problem with step-deteriorating jobs. Journal of Industrial and Management Optimization, 10, 1071-1090.

[11] Hardy, G.H., Littlewood, J.E., and Polya, G. (1976). Inequalities. Cambridge: Cambridge University Press.

[12] Hoogeveen, H. (2005). Multicriteria scheduling. European Journal of Operational Research, 167, 592-623.

[13] Huang, X., Wang, M.-Z., and Ji, P. (2014). Parallel machines scheduling with deteriorating and learning effects. Optimization Letters, 8, 493-500.

[14] Kacem, I., and Levner, E. (2016). An improved approximation scheme for scheduling a maintenance and proportional deteriorating jobs. Journal of Industrial and Management Optimization, 12, 811-817.

[15] Kanet, J.J. (1981). Minimizing variation of flow time in single machine systems. Management Science, 27, 1453-1459.

[16] Li, G., Luo, M.-L., Zhang, W.-J., and Wang, X.-Y. (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.

[17] Li, L., Yang, S.-W., Wu, Y.-B., Huo, Y., and Ji, P. (2013). Single machine scheduling jobs with a truncated sum-of-processing-times-based learning effect. International Journal of Advanced Manufacturing Technology, 67, 261-267.

[18] Low, C., and Lin, W.-Y. (2013). Some scheduling problems with time-dependent learning effect and deteriorating jobs. Applied Mathematical Modelling, 37, 8865-8875.

[19] Mosheiov, G., and Sidney, J.B. (2003). Scheduling with general job-dependent learning curves. European Journal of Operational Research, 147, 665-670.

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

[21] Qian, J., and Steiner, G. (2013). Fast algorithms for scheduling with learning effects and time-dependent processing times on a single machine. European Journal of Operational Research, 225, 547-551.

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

[23] Sun, L.-H., Cui, K., Chen, J.-H., Wang, J., and He, X.-C. (2013). Some results of the worst-case analysis for flow shop scheduling with a learning effect. Annals of Operations Research, 211, 481-490.

[24] Wang, J.-B., and Wang, M.-Z. (2013a). Minimizing makespan in three-machine flow shops with deteriorating jobs. Computers & Operations Research, 40, 547-557.

[25] Wang, X.-R., and Wang, J.-J. (2013b). Single-machine scheduling with convex resource dependent processing times and deteriorating jobs. Applied Mathematical Modelling, 37, 2388-2393.

[26] Wang, X.-Y. and Wang, J.-J. (2013c). Single-machine due date assignment problem with deteriorating jobs and resource-dependent processing times. International Journal of Advanced Manufacturing Technology, 67, 255-260.

[27] Wang, J.-B., Wang, M.-Z., and Ji, P. (2012). Scheduling jobs with processing times dependent on position, starting time and allotted resource. Asia-Pacific Journal of Operational Research, 29(5), 1250030 (15 pages).

[28] Wang, X.-R., Wang, J.-B., Jin, J., and Ji, P. (2014). Single machine scheduling with truncated job-dependent learning effect. Optimization Letters, 8, 669-677.

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

[30] Wang, J.-B., Wang, X.-Y., Sun, L.-H., and Sun, L.-Y. (2013a). Scheduling jobs with truncated exponential learning functions. Optimization Letters, 7(8), 1857-1873.

[31] Wang, X.-Y., Zhou, Z., Zhang, X., Ji, P., and Wang, J.-B. (2013b). Several flow shop scheduling problems with truncated position-based learning effect. Computers & Operations Research, 40, 2906-2929.

[32] Wei, C.-M., Wang, J.-B., and Ji, P. (2012). Single-machine scheduling with time-and-resource-dependent processing times. Applied Mathematical Modelling, 62, 792-798.

[33] Wu, C.-C., Yin,Y., and Cheng, S.-R. (2011). Some single-machine scheduling problems with a truncation learning effect. Computers & Industrial Engineering, 60, 790-795.

[34] Wu, C.-C., Yin,Y., and Cheng, S.-R. (2013). Single-machine and two-machine flowshop scheduling problems with truncated position-based learning functions. Journal of the Operation Research Society, 64, 147-156.

[35] Wu, C.-C., Yin, Y., Wu, W.-H., and Cheng, S.-R. (2012). Some polynomial solvable single-machine scheduling problems with a truncation sum-of-processing-times based learning effect. European Journal of Industrial Engineering, 6, 441-453.

[36] Wu, W.-H., Yin, Y., Wu, W.-H., Wu, C.-C., and Hsu, P.-H. (2014). A time-dependent scheduling problem to minimize the sum of the total weighted tardiness among two agents. Journal of Industrial and Management Optimization, 10, 591-611.

[37] Xu, D., Sun, K., and Li, H. (2008). Parallel machine scheduling with almost periodic maintenance and non-preemptive jobs to minimize makespan. Computers & Operations Research, 35, 1344-1349.

[38] Xu, D., Wan, L., Liu, A., and Yang, D.-L. (2015). Single machine total completion time scheduling problem with workload-dependent maintenance duration. Omega-The International Journal of Management Science, 52, 101-106.

[39] Yang, S.-J. (2013). Unrelated parallel-machine scheduling with deterioration effects and deteriorating multi-maintenance activities for minimizing the total completion time. Applied Mathematical Modelling 37, 2995-3005.

[40] Yang, D.-L., and Kuo, W.-H. (2010). Some scheduling problems with deteriorating jobs and learning effects. Computers & Industrial Engineering 58, 25-28.

[41] Yang, D.-L., Cheng, T.C.E., and Yang, S.-J. (2014a). Parallel-machine scheduling with controllable processing times and rate-modifying activities to minimise total cost involving total completion time and job compressions. International Journal of Production Research, 52, 1133-1141.

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

[43] Yin, Y., Cheng, T.C.E., Wu, C.-C., and Cheng, S.-R. (2014a). Single-machine due window assignment and scheduling with a common flow allowance and controllable job processing time. Journal of the Operational Research Society, 65, 1-13.

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