J. Appl. Math. Comput. (2018) 56:715–725
https://doi.org/10.1007/s12190-017-1091-6

ORIGINAL RESEARCH

Single machine due window assignment resource allocation scheduling with job-dependent learning effect

Na Yin[1]

Received: 22 January 2017 / Published online: 2 March 2017

Abstract This paper deals with a single machine common due window assignment resource allocation scheduling problem with job-dependent learning effect. The objective is to find the due window starting time, a due window size, resource allocation and a job schedule such that total resource consumption cost is minimized subject to a cost function associated with the window location, window size, earliness, tardiness and makespan is less than or equal to a fixed constant number. We show that the problem can be solved in polynomial time. Some extensions of the problem are also given.

Keywords Scheduling · Single-machine · Due-window · Resource allocation · Learning effect

Mathematics Subject Classification 90B35 · 68M20

1 Introduction

We consider the following optimization problem. A set of n jobs $J = \{J_1, J_2, \ldots, J_n\}$ has to be processed on a single machine, and all the jobs are available for processing at time zero. The machine can handle at most one job at a time and job preemption is not allowed. The actual processing time of job J_j when executed in the rth position in a sequence is

$$p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k, \quad u_j > 0, \tag{1}$$

✉ Na Yin
shenyinyin@126.com

[1] School of Science, Shenyang Aerospace University, Shenyang 110136, China

where k is a positive constant, p_j is the normal processing time of job J_j, $a_j \leq 0$ is a position-dependent learning index of job J_j, and u_j is the amount of resource that can be allocated to job J_j. Each job J_j has a unique due window $[d_j^1, d_j^2]$ with $d_j^1 \leq d_j^2$. In this paper we consider a common due window, that is $d_j^1 = d^1$, $d_j^2 = d^2$. Note that the window size, denoted by $D = d^2 - d^1$, is identical for all jobs.

For a given schedule S, let $C_j = C_j(S)$ denote the completion time of job J_j, $C_{\max} = \max\{C_j | j = 1, 2, \ldots, n\}$ be the makespan, $E_j = \max\{0, d^1 - C_j\}$ be the earliness value of job J_j, $T_j = \max\{0, C_j - d^2\}$ be the tardiness value of job J_j, $j = 1, 2, \ldots, n$. The objective is to determine (i) a job schedule S, (ii) a resource allocation $\mathbf{u} = (u_1, u_2, \ldots, u_n)$, (iii) a due window starting time d^1, and (iv) a due window size D such that the following objective function is minimized

$$\sum_{j=1}^{n} G_j u_j, \tag{2}$$

subject to $\sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d^1 + \gamma D) + \eta C_{\max} \leq C$, where G_j is the per time unit cost associated with the resource allocation and $C > 0$ is a given constant. Using the three-field notation of Graham et al. [2], Biskup [1] and Shabtay and Steiner [11], the problem can be denoted as $1\left|p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d^1 + \gamma D)\right.$ $\left. + \eta C_{\max} \leq C\right| \sum_{j=1}^{n} G_j u_j$.

As far as we know, some resource allocation scheduling problems with learning effect has been considered in the literature. Wang et al. [16] considered the single machine scheduling problems $1\left|p_j^A = \left(\frac{p_j r^a}{u_j}\right)^k\right| \delta_1 C_{\max} + \delta_2 TC + \delta_3 TADC +$ $\sum_{j=1}^{n} G_j u_j$ and $1\left|p_j^A = \left(\frac{p_j r^a}{u_j}\right)^k\right| \delta_1 C_{\max} + \delta_2 TW + \delta_3 TADW + \sum_{j=1}^{n} G_j u_j$, where $TC = \sum_{j=1}^{n} C_j$ ($TW = \sum_{j=1}^{n} W_j$) is the total completion time (total waiting time), $TADC$ ($TADW$) is the total absolute differences in completion times (total absolute differences in waiting times), and $W_j = C_j - p_j^A$ is the waiting time of job J_j. They proved that these two problems can be solved in polynomial time. Lu et al. [9] considered the problem $1\left|p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k\right| \sum_{j=1}^{n}(\alpha E_j +$ $\beta T_j + \delta d_j) + \sum_{j=1}^{n} G_j u_j$, where $E_j = \max\{0, d_j - C_j\}$) is the earliness value of job J_j, $T_j = \max\{0, C_j - d_j\}$ is the tardiness value of job J_j, $j = 1, 2, \ldots, n$. For two due date assignment methods (include the common (CON) due date (i.e., $d_j = d$ for all jobs), and the slack (SLK) due date (i.e., $d_j = p_j^A + q$)), they proved that the problem can be solved in polynomial time. Wang and Wang [14] considered the problems $1\left|p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^{n} u_j \leq U\right| \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d_j)$ and $1\left|p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d_j) \leq R\right| \sum_{j=1}^{n} u_j$, where $U > 0$ and $R > 0$ are given constants. For three due date assignment methods (include the CON due date, the SLK due date, and unrestricted (DIF) due date assignment method), they proved that these problems can be solved in polynomial time. Wang and Wang [14] also proved that some scheduling problems without due dates (i.e., $1\left|p_j^A=\left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n u_j \leq U\right|\delta_1 C_{\max}+\delta_2 TC+\delta_3 TADC$, $1\left|p_j^A=\left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n u_j \leq U\right|\delta_1 C_{\max}+\delta_2 TW+\delta_3 TADW$, $1\left|p_j^A=\left(\frac{p_j r^{a_j}}{u_j}\right)^k, \delta_1 C_{\max}+\delta_2 TC+\delta_3 TADC \leq R\right|\sum_{j=1}^n u_j$ and $1\left|p_j^A=\left(\frac{p_j r^{a_j}}{u_j}\right)^k, \delta_1 C_{\max}+\delta_2 TW+\delta_3 TADW \leq R\right|\sum_{j=1}^n u_j$) can be solved in polynomial time. Wang and Wang [13] considered single machine common due-window scheduling problem. They proved that the problem $1\left|p_j^A=\left(\frac{p_j r^{a_j}}{u_j}\right)^k\right|$ $\sum_{j=1}^n(\alpha E_j+\beta T_j+\delta d^1+\gamma D)+\theta\sum_{j=1}^n G_j u_j$ can be solved in polynomial time, where α, β, δ and γ be the per time unit penalties for earliness, tardiness, due date and due window size, respectively. Yang et al. [19] considered single machine resource allocation scheduling problems with multiple due windows. For a non-regular objective cost, they proved that the problem can be solved in polynomial time. Li et al. [5] considered the slack due window scheduling problem $1\left|p_j=\left(\frac{p_j r^{a_j}}{u_j}\right)^k\right|\sum_{j=1}^n(\alpha E_j+\beta T_j+\delta d_j^1+\gamma D_j)+\eta C_{\max}+\theta\sum_{j=1}^n G_j u_j$ can be solved in polynomial time, where $[d_j^1=p_j+q^1, d_j^2=p_j+q^2]$ is the due-window of job J_j, D_j is due-window size, both q^1 and q^2 are decision variables. Li et al. [5] also proved that the problems $1\left|p_j^A=\left(\frac{p_j r^{a_j}}{u_j}\right)^k\right|\delta_1 C_{\max}+\delta_2 TC+\delta_3 TADC+\sum_{j=1}^n G_j u_j$ and $1\left|p_j^A=\left(\frac{p_j r^{a_j}}{u_j}\right)^k\right|\delta_1 C_{\max}+\delta_2 TW+\delta_3 TADW+\sum_{j=1}^n G_j u_j$ can be solved in polynomial time.

The recent paper "Study on due-window scheduling with controllable processing times and learning effect" Wang et al. [12] considered single machine common due window scheduling with limited resource cost availability constraint, i.e., the problem $1\left|p_j^A=\left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n G_j u_j \leq V\right|\sum_{j=1}^n(\alpha E_j+\beta T_j+\delta d+\gamma D)$. They proved that this problem can be solved in polynomial time. In this paper, we study the "inverse version" of the problem studied by Wang et al. [12], that is the case that processing time of a job is described by a convex decreasing resource consumption function and a decreasing position dependent function, and the objective is to minimize the total resource consumed cost subject to a constraint on $\sum_{j=1}^n(\alpha E_j+\beta T_j+\delta d^1+\gamma D)+\eta C_{\max}\leq C$, i.e., the problem $1\left|p_j^A=\left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n(\alpha E_j+\beta T_j+\delta d^1+\gamma D)+\eta C_{\max}\leq C\right|\sum_{j=1}^n G_j u_j$. For more details on scheduling with learning effects, controllable processing times and due windows, the reader may refer to the recent surveys by Biskup [1], Shabtay and Steiner [11] and Janiak et al. [4].

The remainder of this paper is organized as follows. Section 2 derives the properties of the optimal schedule and provides solution algorithm for the general case of the problem. Section 3 considers a special case of the problem, i.e., $a_j = a$ for all jobs. We extend the problem to incorporate with the job-dependent penalty problem, the truncated job-dependent learning effect and the slack due window assignment method in Sect. 4. The last section contains some conclusions and suggests some future research topics.

2 The single machine problem

2.1 Optimal resource allocation

For a given feasible resource allocation $\mathbf{u}$, which fixes the job processing times and the resource consumption cost, our problem reduces to find (i) a job schedule S, (ii) a due window starting time d^1, and (iii) a due window size D such that the objective function $\sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d^1 + \gamma D) + \eta C_{\max}$ is minimized. Similarly to Liman et al. [6,7], Yin et al. [20], Liu et al. [8], we have

Theorem 1 *For problem* $1 \left| p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k \right| \sum_{j=1}^{n}(\alpha E_j+\beta T_j+\delta d^1+\gamma D)+\eta C_{\max}$, *an optimal schedule S satisfies the following properties:*

(1) *All the jobs are processed consecutively without any machine idle from time zero.*
(2) *The optimal values $d^1 = C_{[h]}$ and $d^2 = C_{[l]}$ ($l \geq h$), where $h = \lceil n(\gamma - \delta)/\alpha \rceil$, $l = \lceil n(\beta - \gamma)/\beta \rceil$ and $[j]$ denotes the jth job in a sequence.*

Now, we consider the following cost component:

(1) The earliness cost for job $J_{[j]}$ ($j = 1, 2, \ldots, h$) is:

$$\alpha \sum_{j=1}^{n} E_j = \alpha \sum_{j=1}^{h}(d^1 - C_{[j]}) = \alpha \sum_{j=1}^{h}(C_{[h]} - C_{[j]}) = \alpha \sum_{j=1}^{k}(j-1)p_{[j]}^A$$

(2) The tardiness cost for job $J_{[j]}$ ($j = l+1, l+2, \ldots, n$) is:

$$\beta \sum_{j=1}^{n} T_j = \beta \sum_{j=l+1}^{n}(C_{[j]} - d^2) = \beta \sum_{j=l+1}^{n}(C_{[j]} - C_{[l]}) = \beta \sum_{j=l+1}^{n}(n-j+1)p_{[j]}^A$$

(3) $\delta \sum_{j=1}^{n} d^1 = \delta n d^1 = \sum_{j=1}^{h} \delta n p_{[j]}^A$
(4) $\gamma \sum_{j=1}^{n} D = n\gamma D = n\gamma(C_{[l]} - C_{[k]}) = n\gamma \sum_{j=h+1}^{l} p_{[j]}^A$
(5) $\eta C_{\max} = \eta \sum_{j=1}^{n} p_{[j]}^A$

Hence, from (1), Theorem 2.1 and (1)–(5), we have

$$\begin{aligned}\sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d^1 + \gamma D) + \eta C_{\max} &= \sum_{j=1}^{n} W_j p_{[j]}^A \\ &= \sum_{j=1}^{n} W_j \left(\frac{p_{[j]} j^{a_{[j]}}}{u_{[j]}}\right)^k, \end{aligned} \tag{3}$$

where

$$W_j = \begin{cases} \delta n + \alpha(j-1) + \eta & \text{for } j = 1, 2, \ldots, h; \\ n\gamma + \eta & \text{for } j = h+1, h+2, \ldots, l; \\ \beta(n-j+1) + \eta & \text{for } j = l+1, l+2, \ldots, n. \end{cases} \quad (4)$$

Theorem 2 *For a given schedule* $S = (J_{[1]}, J_{[2]}, \ldots, J_{[n]})$, *the optimal resource allocation of the problem* $1\left|p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n(\alpha E_j + \beta T_j + \delta d^1 + \gamma D) + \eta C_{\max} \leq C\right| \sum_{j=1}^n G_j u_j$ *as a function of the job sequence, that is*

$$u_{[j]}^*(\pi) = \frac{(W_j)^{1/(k+1)}\left(p_{[j]} j^{a_{[j]}}\right)^{k/(k+1)}\left(\sum_{j=1}^n (W_j)^{1/(k+1)}\left(G_{[j]}\right)^{k/(k+1)}\left(p_{[j]} j^{a_{[j]}}\right)^{k/(k+1)}\right)^{1/k}}{C^{1/k}(G_{[j]})^{1/(k+1)}},$$
$$j = 1, 2, \ldots, n, \quad (5)$$

where W_j *is calculated by* (4).

Proof For any given sequence $S = (J_{[1]}, J_{[2]}, \ldots, J_{[n]})$, the Lagrange function is

$$\begin{aligned} L(d^1, D, \mathbf{u}, \lambda) &= \sum_{j=1}^n G_j u_j + \lambda\left(\sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d^1 + \gamma D) + \eta C_{\max} - C\right) \\ &= \sum_{j=1}^n G_{[j]} u_{[j]} + \lambda\left(\sum_{j=1}^n W_j\left(\frac{p_{[j]} j^{a_{[j]}}}{u_{[j]}}\right)^k - C\right) \end{aligned} \quad (6)$$

where λ is the Lagrangian multiplier. Deriving (6) with respect to $u_{[j]}$ and λ, we have

$$\frac{\partial L(d^1, D, \mathbf{u}, \lambda)}{\partial \lambda} = \sum_{j=1}^n W_j\left(\frac{p_{[j]} j^{a_{[j]}}}{u_{[j]}}\right)^k - C = 0, \quad (7)$$

$$\frac{\partial L(d^1, D, \mathbf{u}, \lambda)}{\partial u_{[j]}} = G_{[j]} - k\lambda W_j \times \frac{\left(p_{[j]} j^{a_{[j]}}\right)^k}{(u_{[j]})^{k+1}} = 0. \quad (8)$$

Using (7) and (8) we have

$$u_{[j]} = \frac{\left(k\lambda W_j\left(p_{[j]} j^{a_{[j]}}\right)^k\right)^{1/(k+1)}}{(G_{[j]})^{1/(k+1)}}, \quad (9)$$

$$(k\lambda)^{k/(k+1)} = \frac{\sum_{j=1}^n (W_j)^{1/(k+1)}\left(p_{[j]} G_{[j]} j^{a_{[j]}}\right)^{k/(k+1)}}{C} \quad (10)$$

and

$$u^*_{[j]}(\pi) = \frac{(W_j)^{1/(k+1)}\left(p_{[j]}j^{a_{[j]}}\right)^{k/(k+1)}\left(\sum_{j=1}^{n}(W_j)^{1/(k+1)}\left(G_{[j]}\right)^{k/(k+1)}\left(p_{[j]}j^{a_{[j]}}\right)^{k/(k+1)}\right)^{1/k}}{C^{1/k}(G_{[j]})^{1/(k+1)}}.$$

□

2.2 Optimal sequences

Theorem 3 *For the problem* $1\left|p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n(\alpha E_j + \beta T_j + \delta d^1 + \gamma D) + \eta C_{\max} \le C\right|\sum_{j=1}^n G_j u_j$, *the optimal schedule can be determined by solving a linear assignment problem.*

Proof By substituting (5) into $\sum_{j=1}^n G_j u_j$, we have

$$\sum_{j=1}^{n} G_j u_j(d^1, D, \mathbf{u}, S) = C^{-1/k}\left(\sum_{j=1}^{n}(W_j)^{1/(k+1)}\left(G_{[j]}p_{[j]}j^{a_{[j]}}\right)^{k/(k+1)}\right)^{1+1/k}, \tag{11}$$

W_j is calculated by (4). Let X_{jr} $(j = 1, 2, \ldots, n; r = 1, 2, \ldots, n)$ be a 0–1 variable such that

$$X_{jr} = \begin{cases} 1 & \text{if job } J_j \text{ is processed in the } r\text{th position,} \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

and

$$\theta_{jr} = (W_r)^{1/(k+1)}\left(p_j G_j r^{a_j}\right)^{k/(k+1)}, \tag{13}$$

where

$$W_r = \begin{cases} \delta n + \alpha(r-1) + \eta & \text{for } r = 1, 2, \ldots, h; \\ n\gamma + \eta & \text{for } r = h+1, h+2, \ldots, l; \\ \beta(n-r+1) + \eta & \text{for } r = l+1, l+2, \ldots, n. \end{cases} \tag{14}$$

For k and C are given constant numbers, the optimal schedule of the problem $1\left|p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n(\alpha E_j + \beta T_j + \delta d^1 + \gamma D) + \eta C_{\max} \le C\right|\sum_{j=1}^n G_j u_j$ can be formulated as the following linear assignment problem:

$$\mathbf{P}: \mathrm{Min} Z = \sum_{j=1}^{n}\sum_{r=1}^{n}\theta_{jr}X_{jr} \tag{15}$$

s.t.

$$\sum_{r=1}^{n} X_{jr} = 1, \quad j = 1, 2, \ldots, n, \tag{16}$$

$$\sum_{j=1}^{n} X_{jr} = 1, \quad r = 1, 2, \ldots, n, \tag{17}$$

$$X_{jr} = 0 \text{ or } 1, \quad j, r = 1, 2, \ldots, n. \tag{18}$$

The first set of constraints (Eq. (16)) assures that each job will be assigned only to one position, the second set of constraints (Eq. (17)) assures that each position in the sequence will be occupied by exactly one job. □

2.3 Optimal solution

For $1\left|p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d^1 + \gamma D) + \eta C_{\max} \le C\right| \sum_{j=1}^{n} G_j u_j$ problem, the following optimization algorithm can be proposed.

Algorithm 2.1

Step 1. Calculate the indices h^* and l^*: $h^* = \lceil n(\gamma - \delta)/\alpha \rceil$ and $l^* = \lceil n(\beta - \gamma)/\beta \rceil$.
Step 2. Compute $\theta_{jr} = (W_r)^{1/(k+1)} \left(p_j G_j r^{a_j}\right)^{k/(k+1)}$, where

$$W_r = \begin{cases} \delta n + \alpha(r-1) + \eta & \text{for } r = 1, 2, \ldots, h; \\ n\gamma + \eta & \text{for } r = h+1, h+2, \ldots, l; \\ \beta(n - r + 1) + \eta & \text{for } r = l+1, l+2, \ldots, n. \end{cases}$$

Step 3. Solve the linear assignment problem **P** (i.e., Eqs. (15)–(18)) to determine the optimal schedule S^*.
Step 4. Compute the optimal resources by Eq. (5).
Step 5. Compute the optimal processing times by Eq. (1).
Step 6. Set $d^{1*} = C_{[h^*]}$ and $D^* = C_{[l^*]} - C_{[h^*]}$.

Theorem 4 *The problem* $1\left|p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d^1 + \gamma D) + \eta C_{\max} \le C\right| \sum_{j=1}^{n} G_j u_j$ *can be solved in* $O(n^3)$ *time by Algorithm 2.1.*

Proof The correctness of the algorithm follows from Theorems 2.1, 2.2 and 2.3. It is well known that the linear assignment problem can be solved in $O(n^3)$ time (i.e., Step 3 requires $O(n^3)$ time); Step 2 requires $O(n^2)$; Steps 1, 4, 5, and 6 can be performed in linear time. Thus the overall computational complexity of Algorithm 2.1 is $O(n^3)$. □

In order to illustrate Algorithm 2.1 for the problem $1\left|p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d^1 + \gamma D) + \eta C_{\max} \le C\right| \sum_{j=1}^{n} G_j u_j$, we will solve the following instance:

Example 2.1 Data: $n = 7, k = 1.5, \alpha = 10, \beta = 17, \delta = 4, \gamma = 6, \eta = 1, C = 300$, and the other data are given in Table 1.

Table 1 The data of Example 2.1

J_j	J_1	J_2	J_3	J_4	J_5	J_6	J_7
p_j	35	25	28	19	16	18	12
a_j	−0.05	−0.25	−0.16	−0.20	−0.22	−0.26	−0.15
G_j	5	3	6	4	9	2	8

Table 2 The θ_{jr} values of Example 2.1

	$j\backslash r$	1	2	3	4	5	6	7
$\theta_{jr} =$	1	**85.2672**	94.0195	96.5821	95.7521	95.1133	87.1175	66.4628
	2	51.2857	52.0365	50.9162	48.7658	**47.1605**	42.2611	31.6505
	3	83.2041	87.6421	87.6535	85.2659	83.4587	75.5284	**57.0381**
	4	51.6949	53.5538	53.0421	**51.2423**	49.8883	44.9508	33.8209
	5	75.8537	77.9305	76.8113	73.9491	71.8027	**64.5548**	48.4812
	6	33.0172	33.3616	**32.5640**	31.1349	30.0697	26.9164	20.1398
	7	59.4733	**62.9066**	63.0680	61.4561	60.2342	54.5703	41.2490

Solution:

Step 1. By Theorem 2.1, we have $h^* = \lceil n(\gamma - \delta)/\alpha \rceil = \lceil 7(6-4)/10 \rceil = 2$ and $l^* = \lceil n(\beta - \gamma)/\beta \rceil = \lceil 7(17-6)/17 \rceil = 5$.

Step 2. $W_1 = 29$, $W_2 = 39$, $W_3 = W_4 = W_5 = 43$, $W_6 = 35$, $W_7 = 18$. The values $\theta_{jr} = (W_r)^{1/(k+1)} \left(p_j G_j r^{a_j}\right)^{k/(k+1)}$ are given in Table 2.

Step 3. Solve the linear assignment problem **P** (i.e., Eqs. (15)–(18)), we obtain that $S^* = (J_1 \to J_7 \to J_6 \to J_4 \to J_2 \to J_5 \to J_3)$ (see bold in Table 2).

Step 4. From Eq. (5), we have

$$u_1^* = \frac{(W_j)^{1/(k+1)} \left(p_{[j]} j^{a_{[j]}}\right)^{k/(k+1)} \left(\sum_{j=1}^n (W_j)^{1/(k+1)} \left(G_{[j]}\right)^{k/(k+1)} \left(p_{[j]} j^{a_{[j]}}\right)^{k/(k+1)}\right)^{1/k}}{C^{1/k} (G_{[j]})^{1/(k+1)}}$$
$$= 13.4661,\ u_7^* = 5.4778,$$
$$u_6^* = 5.7616, u_4^* = 10.7360, u_2^* = 14.2249, u_5^* = 4.8422, u_3^* = 7.1504.$$

Step 5. From Eq. (1), we have $p_1^A = (35 \times 1^{(-0.05)}/13.4661)^{1.5} = 4.1902$, $p_7^A = 2.7742$, $p_6^A = 3.5976$, $p_4^A = 1.5533$, $p_2^A = 1.2742$, $p_5^A = 3.3253$, $p_3^A = 4.8576$.

Step 6. Set $d^* = C_{[2^*]} = 4.1902 + 2.7742 = 6.9644$ and $D^* = C_{[h^*]} - C_{[l^*]} = 3.5976 + 1.5533 + 1.2742 = 6.4251$.

3 A special case

If $a_j = a$ for all jobs, stem from (11), we have

$$\sum_{j=1}^{n} G_j u_j = C^{-1/k}\left(\sum_{j=1}^{n}(W_j)^{1/(k+1)}\left(G_{[j]}p_{[j]}j^a\right)^{k/(k+1)}\right)^{1+1/k}$$
$$= C^{-1/k}\left(\sum_{j=1}^{n}\mu_j \nu_{[j]}\right)^{1+1/k}, \tag{19}$$

where

$$\mu_j = (W_j)^{1/(k+1)}(j)^{ak/(k+1)}, \tag{20}$$
$$\nu_{[j]} = \left(p_{[j]}G_{[j]}\right)^{k/(k+1)}, \tag{21}$$

where W_j is calculated by Eq. (4).

Theorem 5 *Problem* $1\left|p_j^A = \left(\frac{p_j r^a}{u_j}\right)^k, \sum_{j=1}^n(\alpha E_j + \beta T_j + \delta d^1 + \gamma D) + \eta C_{\max} \le C\right|\sum_{j=1}^n G_j u_j$ *can be solved in* $O(n\log n)$ *time.*

Proof An optimal solution to the problem $1\left|p_j^A = \left(\frac{p_j r^a}{u_j}\right)^k, \sum_{j=1}^n(\alpha E_j + \beta T_j + \delta d^1 + \gamma D) + \eta C_{\max} \le C\right|\sum_{j=1}^n G_j u_j$ can be constructed as follows: calculate the indices h^* and l^* (according Theorem 2.1), and then calculate $\mu_j = (W_j)^{1/(k+1)}(j)^{ak/(k+1)}$. Assign the smallest μ_j to the job with the largest $\nu_j = \left(p_j G_j\right)^{k/(k+1)}$, the second smallest μ_j to the job with the second largest ν_j, and so on. This matching procedure requires $O(n\log n)$ time (Hardy et al. [3]). Denote the optimal sequence determined in this way by $S^* = (J_{[1]}, J_{[2]}, \ldots, J_{[n]})$ and calculate the optimal resources by Eq. (5), the optimal processing times by Eq. (1), and $d^{1*} = C_{[h^*]}$ and $D^* = C_{[l^*]} - C_{[h^*]}$. □

4 Extensions

4.1 Extension 1

Similar to Sect. 3, the proposed model can be extended to the slack due window scheduling problem (Li et al. [5]) $1\left|p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n(\alpha E_j + \beta T_j + \delta d_j^1 + \gamma D_j) + \eta C_{\max} \le C\right|\sum_{j=1}^n G_j u_j$, where $[d_j^1 = p_j + q^1, d_j^2 = p_j + q^2]$ is the due-window of job J_j, $D_j = d_j^2 - d_j^1 = q^2 - q^1$ is due-window size, both q^1 and q^2 are decision variables.

Theorem 6 *The problem* $1\left|p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n(\alpha E_j + \beta T_j + \delta d_j^1 + \gamma D_j) + \eta C_{\max} \leq C\right| \sum_{j=1}^n G_j u_j$ *can be solved in* $O(n^3)$ *time.*

Theorem 7 *Problem* $1\left|p_j^A = \left(\frac{p_j r^{a}}{u_j}\right)^k, \sum_{j=1}^n(\alpha E_j + \beta T_j + \delta d_j^1 + \gamma D_j) + \eta C_{\max} \leq C\right| \sum_{j=1}^n G_j u_j$ *can be solved in* $O(n \log n)$ *time.*

4.2 Extension 2

Similar to Sect. 3, the proposed model can be extended to a large set of scheduling problems, i.e., $1\left|p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n W_j p_{[j]}^A \leq C\right| \sum_{j=1}^n G_j u_j$, where W_j is a position, job-dependent penalty for any job schedule in the jth position.

Theorem 8 *The problem* $1\left|p_j^A = \left(\frac{p_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n W_j p_{[j]}^A \leq C\right| \sum_{j=1}^n G_j u_j$ *can be solved in* $O(n^3)$ *time.*

Theorem 9 *Problem* $1\left|p_j^A = \left(\frac{p_j r^{a}}{u_j}\right)^k, \sum_{j=1}^n W_j p_{[j]}^A \leq C\right| \sum_{j=1}^n G_j u_j$ *can be solved in* $O(n \log n)$ *time.*

4.3 Extension 3

In the real production process, learning effect may not reduce the jobs' processing time without limitation (Wu et al. [17,18]), hence Wang et al. [15] considered the truncated job-dependent learning effect model, i.e., $p_j^A = p_j \max\{r^{a_j}, B\}$, where $0 < B \leq 1$ is a truncation parameter for all jobs. Similar to the proof of Sect. 3, the proposed model can be extended to the following model: $p_j^A = \left(\frac{p_j \max\{r^{a_j}, B\}}{u_j}\right)^k$.

Theorem 10 *The problem* $1\left|p_j^A = \left(\frac{p_j \max\{r^{a_j}, B\}}{u_j}\right)^k, \sum_{j=1}^n W_j p_{[j]}^A \leq C\right| \sum_{j=1}^n G_j u_j$ *can be solved in* $O(n^3)$ *time.*

Theorem 11 *Problem* $1\left|p_j^A = \left(\frac{p_j \max\{r^{a}, B\}}{u_j}\right)^k, \sum_{j=1}^n W_j p_{[j]}^A \leq C\right| \sum_{j=1}^n G_j u_j$ *can be solved in* $O(n \log n)$ *time.*

5 Conclusions

In this paper, we have considered the scheduling problem with learning effect and resource-dependent processing times on a single machine. It is showed that the due window assignment minimization problem can be solved in polynomial time. For the special case of the problem, we also gave a lower order algorithm. The algorithms can also be easily applied to the problems with the deterioration (aging) effect (e.g.,

$a_j > 0$). For future research, it is worthwhile to study other scheduling problems with due window assignment, effects of deterioration and truncated job-dependent learning (Niu et al. [10]) and/or resource allocation, for example, the flow shop scheduling problems and other scheduling performance measures.

Acknowledgements This research was supported by the National Natural Science Foundation of China (Grant no. 71471120) and the Program for Liaoning Natural Science Foundation Research (Grant no. 201601177).

References

1. Biskup, D.: A state-of-the-art review on scheduling with learning effects. Eur. J. Oper. Res. **188**, 315–329 (2008)
2. Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.H.G.R.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann. Discrete Math. **5**, 287–326 (1979)
3. Hardy, G.H., Littlewood, J.E., Polya, G.: Inequalities, 2nd edn. Cambridge University Press, Cambridge (1952)
4. Janiak, A., Janiak, W.A., Krysiak, T., Kwiatkowski, T.: A survey on scheduling problems with due windows. Eur. J. Oper. Res. **242**, 347–357 (2015)
5. Li, G., Luo, M.-L., Zhang, W.-J., Wang, X.-Y.: Single-machine due-window assignment scheduling based on common flow allowance, learning effect and resource allocation. Int. J. Prod. Res. **53**, 1228–1241 (2015)
6. Liman, S.D., Panwalkar, S.S., Thongmee, S.: Determination of common due window location in a single machine scheduling problem. Eur. J. Oper. Res. **93**, 68–74 (1996)
7. Liman, S.D., Panwalkar, S.S., Thongmee, S.: Common due window size and location determination in a single machine scheduling problem. J. Oper. Res. Soc. **49**, 1007–1010 (1998)
8. Liu, L., Wang, J.-J., Wang, X.-Y.: Single machine due-window assignment scheduling with resource-dependent processing times to minimize total resource consumption cost. Int. J. Prod. Res. **54**, 1186–1195 (2015)
9. Lu, Y.-Y., Li, G., Wu, Y.-B., Ji, P.: Optimal due-date assignment problem with learning effect and resource-dependent processing times. Optim. Lett. **8**, 113–127 (2014)
10. Niu, Y.-P., Wang, J., Yin, N.: Scheduling problems with effects of deterioration and truncated job-dependent learning. J. Appl. Math. Comput. **47**, 315–325 (2015)
11. Shabtay, D., Steiner, G.: A survey of scheduling with controllable processing time. Discrete Appl. Math. **115**(13), 1643–1666 (2007)
12. Wang, X.-Y., Li, L., Wang, J.-B.: Study on due-window scheduling with controllable processing times and learning effect. ICIC Express Lett. **10**, 2913–2919 (2016)
13. Wang, J.-B., Wang, M.-Z.: Single-machine due-window assignment and scheduling with learning effect and resource-dependent processing times. Asia-Pac. J. Oper. Res. **31**(5), 1450036 (2014)
14. Wang, J.-B., Wang, J.-J.: Research on scheduling with job-dependent learning effect and convex resource dependent processing times. Int. J. Prod. Res. **53**, 5826–5836 (2015)
15. Wang, X.-R., Wang, J.-B., Jin, J., Ji, P.: Single machine scheduling with truncated job-dependent learning effect. Optim. Lett. **8**(2), 669–677 (2014)
16. Wang, D., Wang, M.-Z., Wang, J.-B.: Single-machine scheduling with learning effect and resource-dependent processing times. Comput. Ind. Eng. **59**, 458–462 (2010)
17. Wu, C.-C., Yin, Y., Cheng, S.-R.: Single-machine and two-machine flowshop scheduling problems with truncated position-based learning functions. J. Oper. Res. Soc. **64**, 147–156 (2013)
18. Wu, C.-C., Yin, Y., Wu, W.-H., Cheng, S.-R.: Some polynomial solvable single-machine scheduling problems with a truncation sum-of-processing-times based learning effect. Eur. J. Ind. Eng. **6**, 441–453 (2012)
19. Yang, D.-L., Lai, C.-J., Yang, S.-J.: Scheduling problems with multiple due windows assignment and controllable processing times on a single machine. Int. J. Prod. Econ. **150**, 96–103 (2014)
20. Yin, Y., Cheng, T.C.E., Wu, C.-C., Cheng, S.-R.: Single-machine due window assignment and scheduling with a common flow allowance and controllable job processing time. J. Oper. Res. Soc. **65**, 1–13 (2014)