ICIC Express Letters ICIC International ©2016 ISSN 1881-803X
Volume **10**, Number **12**, December **2016** pp. **2913–2919**

STUDY ON DUE-WINDOW SCHEDULING WITH CONTROLLABLE PROCESSING TIMES AND LEARNING EFFECT

Xiaoyuan Wang, Lin Li* and Jibo Wang

School of Science
Shenyang Aerospace University
No. 37, Daoyi South Avenue, Shenbei New Area, Shenyang 110136, P. R. China
wxy5099@126.com; *Corresponding author: lilinsau@163.com; wangjibo75@163.com

Received May 2016; accepted August 2016

Abstract. *In this paper, we investigate common due-window assignment scheduling problem with controllable processing times and learning effect on a single machine. Our goal is to find the window location and size, along with the associated job schedule to minimize a cost function associated with the window location, window size, earliness and tardiness. We show that the problem can be solved in $O(n^3)$ time. We also show that a special case of the problem can be solved by a lower order algorithm.*

Keywords: Scheduling, Due-window, Controllable processing times, Learning effect

1. **Introduction.** In recent years, by relaxing the classical scheduling assumption, a lot of work has been done on the phenomenon of learning and/or controllable processing time. A survey on this line of the scheduling problems with learning effects (controllable processing times) could be found in Biskup [1] (Shabtay and Steiner [2]). More recent papers which have considered scheduling problems with learning effects and/or controllable processing times include Choi et al. [3], Leyvand et al. [4], Wang et al. [5], Yin and Wang [6], Yin et al. [7], Yin et al. [8], Sun et al. [9], Yang et al. [10], Lu et al. [11], and Wang and Wang [12].

On the other hand, increasing attention has been paid to the due-window assignment scheduling problems (Liman et al. [13,14], Mosheiov and Sarig [15-17], Wang and Wang [18], Wang et al. [19], Wu et al. [20], Yin et al. [21], Yang et al. [22], and Liu et al. [23]), i.e., if a job is finished earlier (later) than its due-window, it has to be stored as inventory, which results in an earliness penalty (it will incur a tardiness penalty as stated in the contract). However, to the best of our knowledge, there exist only a few results concerning due-window scheduling problems with learning effects and controllable processing times simultaneously. Wang and Wang [24] considered a convex resource consumption function model in which the processing time of job J_j is $p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k$, $u_j > 0$, where $\bar{p}_j$, a_j, u_j are the basic (normal) processing time, the learning rate, the amount of resource that can be allocated of job J_j respectively, r is the position of job J_j scheduled in a sequence, and k is a positive constant. Using the extended three-field notation scheme (Graham et al. [25]), Wang and Wang [24] proved that the common due-window scheduling problem $1|p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k \Big| \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D) + \theta \sum_{j=1}^n G_j u_j$ can be solved in polynomial time, where α, β, δ and γ represent the per time unit penalties for earliness, tardiness, due date, and due window size, respectively. Li et al. [26] considered the slack due window scheduling problem $1|p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k \Big| \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d_j^1 + \gamma D_j) + \eta C_{\max} + \theta \sum_{j=1}^n G_j u_j$ 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. In the real production process, the resource cost availability is limited; hence, in this paper, we continue the work of Wang and Wang [24], by considering limited resource cost availability constraint, i.e., the problem $1|p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n G_j u_j \le V| \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D)$.

In Section 2, we state the problem formally. In Sections 3 and 4, we provide an $O(n^3)$-time algorithm for the general case and an $O(n \log n)$-time algorithm for a special case. In Section 5, conclusions are presented.

2. **Problem Formulation.** There is given a single machine and n independent jobs $J = \{J_1, J_2, \ldots, J_n\}$ at time zero. As in Wang and Wang [24] and Li et al. [26], we consider the following convex resource consumption model:

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

where k is a positive constant, p_j ($\bar{p}_j$) is the actual (normal) processing time of job J_j, r is the position of job J_j scheduled in a sequence, $a_j \le 0$ 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.

Let $d(\ge 0)$ ($(d + D), D \ge 0$) represent the starting (finishing) time of the due-window, D is the due-window size, d and D are both decision variables. For a given schedule π, let $C_j = C_j(\pi)$ be the completion time of job J_j, $T_j = \max\{0, C_j - d - D\}$ ($E_j = \max\{0, d - C_j\}$) is the tardiness (earliness) value of job J_j, $j = 1, 2, \ldots, n$. The objective is to determine the optimal due date d, due window size D and to find a schedule π which minimizes

$$Z(d, D, \pi, u) = \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D), \tag{2}$$

subject to $\sum_{j=1}^n G_j u_j \le V$, where G_j is the per time unit cost associated with the resource allocation and $V > 0$ is a given constant. Using the three-field notation of Graham et al. [25] the problem can be denoted as $1|p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n G_j u_j \le V| \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D)$.

3. **The Problem $\mathbf{1|p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n G_j u_j \le V| \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D)}$.** If the processing times are given constants, then the following theorem holds.

Theorem 3.1. (Liman et al. [14]) *For the problem* $1||\sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D)$*, there hold the following properties.*

(1) There exists an optimal schedule π^* without any machine idle time between the starting time of the first job and the completion time of the last job. Furthermore, the first job in the schedule starts at time zero.

(2) There exists an optimal schedule with the property that d and $d + D$ coincide with the completion times of the kth and lth jobs ($l \ge k$), i.e., $k = \lceil n(\gamma - \delta)/\alpha \rceil$ and $l = \lceil n(\beta - \gamma)/\beta \rceil$.

(3) The optimal total cost can be written as: $Z(d, D, \pi, u) = \sum_{j=1}^n \omega_j p_{[j]}$, where $[j]$ denotes the jth job in a sequence, and the positional weight of position j in the schedule is given by

$$\omega_j = \min\{n\delta + (j-1)\alpha, n\gamma, (n + 1 - j)\beta\}, \quad r = 1, 2, \ldots, n. \tag{3}$$

Note that jobs in positions $j = 1, 2, \ldots, k$ will be early jobs, jobs in positions $j = k + 1, \ldots, l$ will be window jobs, jobs in positions $j = l + 1, \ldots, n$ will be tardy jobs (Liman et al. [14]), and the values k and l are independent of the actual processing times and the schedule. Hence, from (1) and Theorem 3.1, (2) can be rewritten as

$$Z(d, D, \pi, u) = \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D)$$

$$= \sum_{j=1}^{n} \omega_j p_{[j]} = \sum_{j=1}^{n} \omega_j \left(\frac{\bar{p}_{[j]} j^{a_{[j]}}}{u_{[j]}} \right)^k, \tag{4}$$

where ω_j is calculated by (3).

Theorem 3.2. *For a given schedule $\pi = (J_{[1]}, J_{[2]}, \ldots, J_{[n]})$, the optimal resource allocation of the problem $1|p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k$, $\sum_{j=1}^n G_j u_j \le V| \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D)$ is a function of the job sequence, that is*

$$u^*_{[j]}(\pi) = \frac{(\omega_j)^{1/(k+1)} (G_{[j]})^{-1/(k+1)} \left(\bar{p}_{[j]} j^{a_{[j]}}\right)^{k/(k+1)}}{\sum_{j=1}^n (\omega_j)^{1/(k+1)} \left(\bar{p}_{[j]} G_{[j]} j^{a_{[j]}}\right)^{k/(k+1)}} \times V, \quad j = 1, 2, \ldots, n, \tag{5}$$

where ω_j is calculated by (3).

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

$$\begin{aligned} L(d, D, u, \vartheta) &= \sum_{j=1}^{n} (\alpha E_j + \beta T_j + \delta d + \gamma D) + \vartheta \left(\sum_{j=1}^{n} G_{[j]} u_{[j]} - V \right) \\ &= \sum_{j=1}^{n} \omega_j \left(\frac{\bar{p}_{[j]} j^{a_{[j]}}}{u_{[j]}} \right)^k + \vartheta \left(\sum_{j=1}^{n} G_{[j]} u_{[j]} - V \right) \end{aligned} \tag{6}$$

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

$$\frac{\partial L(d, D, u, \vartheta)}{\partial \vartheta} = \sum_{j=1}^{n} G_{[j]} u_{[j]} - V = 0, \tag{7}$$

$$\frac{\partial L(d, D, u, \vartheta)}{\partial u_{[j]}} = \vartheta G_{[j]} - k\omega_j \times \frac{\left(\bar{p}_{[j]} j^{a_{[j]}}\right)^k}{(u_{[j]})^{k+1}} = 0. \tag{8}$$

Using (7) and (8) we obtain

$$u_{[j]} = \frac{\left(k\omega_j \left(\bar{p}_{[j]} j^{a_{[j]}}\right)^k\right)^{1/(k+1)}}{(\vartheta G_{[j]})^{1/(k+1)}} \tag{9}$$

and

$$\vartheta^{1/(k+1)} = \frac{\sum_{j=1}^n (k\omega_j)^{1/(k+1)} \left(\bar{p}_{[j]} G_{[j]} j^{a_{[j]}}\right)^{k/(k+1)}}{V}. \tag{10}$$

From (9) and (10), we have

$$u^*_{[j]}(\pi) = \frac{(\omega_j)^{1/(k+1)} (G_{[j]})^{-1/(k+1)} \left(\bar{p}_{[j]} j^{a_{[j]}}\right)^{k/(k+1)}}{\sum_{j=1}^n (\omega_j)^{1/(k+1)} \left(\bar{p}_{[j]} G_{[j]} j^{a_{[j]}}\right)^{k/(k+1)}} \times V.$$

□

Theorem 3.3. *For the problem $1|p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k$, $\sum_{j=1}^n G_j u_j \le V| \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D)$, the optimal schedule can be determined by solving an assignment problem.*

Proof: Substituting (5) into (4), we obtain a new unified expression for the objective function $\sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D)$ under an optimal resource allocation:

$$Z(d, D, \pi, u^*) = V^{-k} \left(\sum_{j=1}^{n} (\omega_j)^{1/(k+1)} \left(\bar{p}_{[j]} G_{[j]} j^{a_{[j]}}\right)^{k/(k+1)} \right)^{k+1}, \tag{11}$$

ω_j is calculated by (3). 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

$$\lambda_{jr} = (\omega_r)^{1/(k+1)} \left(\bar{p}_j G_j r^{a_j}\right)^{k/(k+1)}. \tag{13}$$

Then the optimal schedule of the problem $1|p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n G_j u_j \leq V| \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D)$ can be formulated as the following linear assignment problem:

$$\text{Min} \quad Z = V^{-k} \left(\sum_{j=1}^n \sum_{r=1}^n \lambda_{jr} x_{jr} \right)^{k+1} \tag{14}$$

$$\text{s.t.} \quad \sum_{r=1}^n x_{jr} = 1, \quad j = 1, 2, \ldots, n, \tag{15}$$

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

$$x_{jr} = 0 \text{ or } 1, \quad j, r = 1, 2, \ldots, n. \tag{17}$$

□

For the problem $1|p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n G_j u_j \leq V| \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D)$, we can propose the following optimization algorithm.

Algorithm 3.1
Step 1. Determine the value of $k^* = \lceil n(\gamma - \delta)/\alpha \rceil$ and $l^* = \lceil n(\beta - \gamma)/\beta \rceil$.
Step 2. Compute $\lambda_{jr} = (\omega_r)^{1/(k+1)} \left(\bar{p}_j G_j r^{a_j}\right)^{k/(k+1)}$ where ω_r is calculated by (3).
Step 3. Solve the linear assignment problem (14)-(17) to determine the optimal schedule π^*.
Step 4. Compute the optimal resources by (5).
Step 5. Compute the optimal processing times by (1).
Step 6. Set $d^* = C_{[k^*]}$ and $D^* = C_{[l^*]} - C_{[k^*]}$.

Theorem 3.4. *The scheduling problem* $1|p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n G_j u_j \leq V| \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D)$ *can be solved in* $O(n^3)$ *time by Algorithm 3.1.*

Proof: The correctness of the algorithm follows from Theorems 3.1, 3.2 and 3.3. Step 2 requires $O(n^2)$ and Step 3 $O(n^3)$ time; Steps 1, 4, 5, and 6 can be performed in linear time. Thus the overall computational complexity of Algorithm 3.1 is $O(n^3)$. □

In order to illustrate Algorithm 3.1 for $1|p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^n G_j u_j \leq V| \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D)$, we solve the instance as follows:

Example 3.1. *Data:* $n = 7$, $k = 2$, $\alpha = 11$, $\beta = 18$, $\delta = 5$, $\gamma = 7$, $V = 200$, *and the other data are given in Table 1.*

Table 1. The data of Example 3.1

J_i	J_1	J_2	J_3	J_4	J_5	J_6	J_7
$\bar{p}_j$	25	20	26	18	15	16	10
a_j	-0.05	-0.20	-0.06	-0.23	-0.32	-0.16	-0.15
G_j	5	2	6	3	7	1	8

Solution.
Step 1. By Theorem 3.1, we have $k^* = \lceil n(\gamma - \delta)/\alpha \rceil = \lceil 7(7-5)/11 \rceil = 2$ and $l^* = \lceil n(\beta - \gamma)/\beta \rceil = \lceil 7(18-7)/18 \rceil = 5$.
Step 2. $\omega_1 = 35$, $\omega_2 = 46$, $\omega_3 = \omega_4 = \omega_5 = 49$, $\omega_6 = 36$, $\omega_7 = 18$. The values $\lambda_{jr} = (\omega_r)^{1/(k+1)} \left(\bar{p}_j G_j r^{a_j}\right)^{k/(k+1)}$ are given in Table 2.
Step 3. Solve the linear assignment problem (14)-(17), and we obtain that $\pi^* = (J_1 \to J_6 \to J_2 \to J_7 \to J_4 \to J_5 \to J_3)$ (see bold in Table 2).
Step 4. From (5), we have $u_1^* = \frac{(\omega_1)^{1/3}(G_1)^{-1/3}(\bar{p}_1 1^{a_1})^{2/3}}{\sum_{j=1}^n (\omega_j)^{1/3}\left(\bar{p}_{[j]} G_{[j]} j^{a_{[j]}}\right)^{k/(k+1)}} \times 200 = 9.0795$, $u_6^* = 11.7299$, $u_2^* = 10.2611$, $u_7^* = 4.1042$, $u_4^* = 7.5585$, $u_5^* = 3.9767$, $u_3^* = 6.5855$.
Step 5. From (1), we have $p_1^* = 7.5815$, $p_6^* = 1.4905$, $p_2^* = 2.4481$, $p_7^* = 3.9167$, $p_4^* = 2.7049$, $p_5^* = 4.5198$, $p_3^* = 12.3412$.
Step 6. Set $d^* = C_{[2^*]} = 7.5815 + 1.4905 = 9.0720$ and $D^* = C_{[l^*]} - C_{[k^*]} = 2.4481 + 3.9167 + 2.7049 = 9.0697$.

TABLE 2. The λ_{jr} values of Example 3.1

		$r=1$	$r=2$	$r=3$	$r=4$	$r=5$	$r=6$	$r=7$
$\lambda_{jr} =$	J_1	**81.7767**	87.5303	88.1931	87.3514	86.7041	77.7623	61.4037
	J_2	38.2586	38.2081	**36.9677**	35.5766	34.5337	30.4126	23.6475
	J_3	94.7922	100.9937	101.4838	100.3227	99.4313	89.0686	**70.2592**
	J_4	46.7324	46.0282	44.1743	42.2680	**40.8463**	35.8410	27.7825
	J_5	72.8029	68.7848	64.4277	60.5925	57.7756	**50.1443**	38.5120
	J_6	20.7700	**21.1295**	20.6658	20.0413	19.5699	17.3185	13.5216
	J_7	60.7318	62.0692	60.8714	**59.1452**	57.8400	51.2482	40.0536

4. **A Special Case.** In the following, we consider a special case, i.e., $a_j = a$ for all jobs, and present a simpler and more efficient solution for this case.

Lemma 4.1. (Hardy et al. [27]) *The sum of products $\sum_{j=1}^n \varphi_j \psi_j$ is minimized if sequence $\varphi_1, \varphi_2, \ldots, \varphi_n$ is ordered nondecreasingly and sequence $\psi_1, \psi_2, \ldots, \psi_n$ is ordered nonincreasingly or vice versa, and it is maximized if the sequences are ordered in the same way.*

From (11) and $a_j = a$, we have

$$Z(d, D, \pi, u^*) = V^{-k} \left(\sum_{j=1}^n (\omega_j)^{1/(k+1)} \left(\bar{p}_{[j]} G_{[j]} j^a\right)^{k/(k+1)} \right)^{k+1} = VU^{-k} \left(\sum_{j=1}^n \varphi_j \psi_{[j]} \right)^{k+1}, \tag{18}$$

where

$$\varphi_j = (\omega_j)^{1/(k+1)} (j)^{ak/(k+1)}, \tag{19}$$

$$\psi_{[j]} = \left(\bar{p}_{[j]} G_{[j]}\right)^{k/(k+1)}, \tag{20}$$

where ω_j is calculated by (3).

Based on the above analysis Theorems 3.1, 3.2 and Lemma 4.1, we develop an $O(n \log n)$ algorithm that solves the problem $1|p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k$, $a_j = a$, $\sum_{j=1}^n G_j u_j \leq V| \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D)$.

Algorithm 4.1
Step 1. Determine the value of $k^* = \lceil n(\gamma - \delta)/\alpha \rceil$ and $l^* = \lceil n(\beta - \gamma)/\beta \rceil$.
Step 2. Compute φ_j and $\psi_{[j]}$ by (19) and (20).
Step 3. Apply Lemma 4.1 (i.e., HLP rule) to determining the optimal schedule π^*.

Step 4. Compute the optimal resources by (5).
Step 5. Compute the optimal processing times by (1).
Step 6. Set $d^* = C_{[k^*]}$ and $D^* = C_{[l^*]} - C_{[k^*]}$.

Obviously, Algorithm 4.1 can be solved in $O(n \log n)$ time (Step 1 takes constant time and Steps 2, 4-6 take $O(n)$ time, and Step 3 requires the implementation of the HLP rule, which requires $O(n \log n)$ time).

Based on the above analysis, we have the following theorem.

Theorem 4.1. *The* $1|p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k$, $a_j = a$, $\sum_{j=1}^n G_j u_j \le V| \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d + \gamma D)$ *problem can be solved in* $O(n \log n)$ *time by Algorithm 4.1.*

5. **Conclusions.** We considered the single machine scheduling problem with learning effect and resource-dependent processing times. We showed that the weighted combination of the earliness, tardiness, window location, window size, and resource cost minimization problem can be solved in polynomial time. For future research, it is worthwhile to study other scheduling problems with learning effects and resource allocation, for example, the flow shop scheduling problems and the parallel machine scheduling problems.

Acknowledgements. This research was supported by the Science Research Foundation of Shenyang Aerospace University (Grant No. 201410Y) and the National Natural Science Foundation of China (Grant No. 61403260).

REFERENCES

[1] D. Biskup, A state-of-the-art review on scheduling with learning effects, *European Journal of Operational Research*, vol.188, pp.315-329, 2008.

[2] D. Shabtay and G. Steiner, A survey of scheduling with controllable processing times, *Discrete Applied Mathematics*, vol.155, no.13, pp.1643-1666, 2007.

[3] B.-C. Choi, J. Y.-T. Leung and M. L. Pinedo, Complexity of a scheduling problem with controllable processing times, *Operations Research Letters*, vol.38, pp.123-126, 2010.

[4] Y. Leyvand, D. Shabtay and G. Steiner, A unified approach for scheduling with convex resource consumption functions using positional penalties, *European Journal of Operational Research*, vol.206, pp.301-312, 2010.

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

[6] N. Yin and X.-Y. Wang, Single-machine scheduling with controllable processing times and learning effect, *International Journal of Advanced Manufacturing Technology*, vol.54, pp.743-748, 2011.

[7] Y. Yin, T. C. E. Cheng, S.-R. Cheng and C.-C. Wu, Single-machine batch delivery scheduling with an assignable common due date and controllable processing times, *Computers & Industrial Engineering*, vol.65, pp.652-662, 2013.

[8] Y. Yin, T. C. E. Cheng, C.-C. Wu and S.-R. Cheng, Single-machine common due-date scheduling with batch delivery costs and resource-dependent processing times, *International Journal of Production Research*, vol.51, pp.5083-5099, 2013.

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

[10] D.-L. Yang, T. C. E. Cheng and S.-J. Yang, 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*, vol.52, pp.1133-1141, 2014.

[11] Y.-Y. Lu, G. Li, Y.-B. Wu and P. Ji, Optimal due-date assignment problem with learning effect and resource-dependent processing times, *Optimization Letters*, vol.8, pp.113-127, 2014.

[12] J.-B. Wang and J.-J. Wang, Research on scheduling with job-dependent learning effect and convex resource dependent processing times, *International Journal of Production Research*, vol.53, pp.5826-5836, 2015.

[13] S. D. Liman, S. S. Panwalkar and S. Thongmee, Determination of common due window location in a single machine scheduling problem, *European Journal of Operational Research*, vol.93, pp.68-74, 1996.

[14] S. D. Liman, S. S. Panwalkar and S. Thongmee, Common due window size and location determination in a single machine scheduling problem, *Journal of the Operational Research Society*, vol.49, pp.1007-1010, 1998.

[15] G. Mosheiov and A. Sarig, A due-window assignment problem with position dependent processing times, *Journal of the Operational Research Society*, vol.59, pp.997-1003, 2008.

[16] G. Mosheiov and A. Sarig, Scheduling a maintenance activity and due-window assignment on a single machine, *Computers and Operations Research*, vol.36, pp.2541-2545, 2009.

[17] G. Mosheiov and A. Sarig, Scheduling identical jobs and due-window on uniform machines, *European Journal of Operational Research*, vol.201, pp.712-719, 2010.

[18] J.-B. Wang and C. Wang, Single-machine due-window assignment problem with learning effect and deteriorating jobs, *Applied Mathematical Modelling*, vol.35, pp.4017-4022, 2011.

[19] J.-B. Wang, L. Liu and C. Wang, Single machine SLK/DIF due window assignment problem with learning effect and deteriorating jobs, *Applied Mathematical Modelling*, vol.37, pp.8394-8400, 2013.

[20] Y.-B. Wu, L. Wan and X.-Y. Wang, Study on due-window assignment scheduling based on common flow allowance, *International Journal of Production Economics*, vol.165, pp.155-157, 2015.

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

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

[23] L. Liu, J.-J. Wang and X.-Y. Wang, Single machine due-window assignment scheduling with resource-dependent processing times to minimize total resource consumption cost, *International Journal of Production Research*, vol.54, pp.1186-1195, 2015.

[24] J.-B. Wang and M.-Z. Wang, Single-machine due-window assignment and scheduling with learning effect and resource-dependent processing times, *Asia-Pacific Journal of Operational Research*, vol.31, no.5, 2014.

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

[26] G. Li, M.-L. Luo, W.-J. Zhang and X.-Y. Wang, Single-machine due-window assignment scheduling based on common flow allowance, learning effect and resource allocation, *International Journal of Production Research*, vol.53, pp.1228-1241, 2015.

[27] G. H. Hardy, J. E. Littlewood and G. Polya, *Inequalities*, 2nd Edition, Cambridge University Press, Cambridge, 1952.