International Journal of Applied Mathematics

Volume 30 No. 5 2017, 375-386

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

doi: http://dx.doi.org/10.12732/ijam.v30i5.2

SINGLE MACHINE SLACK DUE-WINDOW SCHEDULING WITH LINEAR RESOURCE ALLOCATION, AGING EFFECT, AND A DETERIORATING RATE-MODIFYING ACTIVITY

Bo Cheng[1 §], Ling Cheng[2]

[1]Department of Applied Mathematics
School of Finance
Guangdong University of Foreign Studies
Guangzhou, 510420, P.R. CHINA

[2]School of Electrical and Information Engineering
University of the Witwatersrand
Private Bag 3, Wits. 2050
Johannesburg, SOUTH AFRICA

Abstract: In this paper, we consider the slack due-window method and investigate single machine scheduling with a deteriorating rate-modifying activity, linear resource allocation and aging effect. The objective is to minimize the total cost caused by the due-window location, the due-window size, the earliness and tardiness with respect to a slack due-window, and resource consumption. We provide a polynomial-time algorithm to solve the corresponding problem.

AMS Subject Classification: 90B35

Key Words: aging effect, due-window, rate-modifying activity, resource allocation, scheduling

1. Introduction

In the studies on operations management, a good customer service usually requires jobs completed as close as possible to their due-dates. A time interval is assigned in the supply contract so that a job finished within this period will

Received: August 8, 2017

[§]Correspondence author

be considered on time and not be penalized. This period is called the due-window of a job (cf. [1, 2]). The due-window assignment methods include common due-window, slack due-window (also called common flow allowance) and others.

Scheduling theory for jobs with changeable job processing times has been developed steadily in the last decade. The single machine common due-window assignment problem with deteriorating jobs and learning effect was studied by [3] and polynomial-time algorithms were given to minimize the total costs for earliness, tardiness, window location and window size. The parallel problem for the slack due-window model was investigated by [4]. Other scheduling problems with the aging effect were studied by [5, 6, 7].

The effect of allocating additional resources on job processing times was also extensively investigated by [8, 9, 10] and others. A linear function taking the amount of allocated resource as the input parameter was proposed to quantify the effect of additional resources on job processing times. For example, in a linear resource consumption model, the actual processing time of a job is determined by

$$\tilde{p} = p - cu, \quad 0 \le u \le \bar{u} < \frac{p}{c}, \tag{1}$$

where $\tilde{p}$ is the actual processing time, p is the non-compressed (normal) processing time, c is the positive compression factor, u is the actual resource allocated to the job, and $\bar{u}$ is the maximum resource that could be allocated to the job. For the linear resource consumption model in this paper, we remove the upper bound on $\bar{u}$ in (1), since the new one without the upper bound is more general.

Machine scheduling with a rate-modifying activity (RMA) was initially investigated by [11]. In this paper, at most one RMA can be scheduled before or after any job except that it is unnecessary to schedule a RMA after the last job. The scheduler decides when to perform the RMA. However, the RMA cannot interrupt any job. It means the RMA can be scheduled either before or after a job.

The combinations of the above-mentioned settings have been further investigated. [12] studied common due-window assignment and scheduling with time-dependent deteriorating jobs and a maintenance activity, and [9, 10] investigated the scheduling problem with resource allocation, aging effect and a deteriorating RMA based on common due-window method. [13] considered the slack due-window assignment and scheduling taking into account variable processing-time jobs and a rate-modifying activity.

In this paper, we discuss the problem of slack due-window assignment and single machine scheduling taking into account linear resource allocation and position-dependent deteriorating jobs with a RMA. To our best knowledge, this problem has not been studied in literatures.

The rest of this paper is organized as follows. A description of the problem under study is given in Section 2. In Section 3 a few important lemmas and properties are presented. In the same section, a polynomial-time solution is given. A numerical example is presented to demonstrate the polynomial-time solution in Section 4. The research is concluded and future study is foreseen in the last section.

2. Model Formulation

In the models considered in this paper, n independent and non-preemptive jobs $J_1, J_2, \dots, J_n$ and at most one RMA are scheduled on a single machine. All the jobs and the RMA can be scheduled at time zero. Let p_j be the normal processing time of job J_j. The predetermined parameters of job J_j are the job-dependent aging factor a_j and the maximum available resource amount $\bar{u}_j$. Let p_{jr} denote the actual processing time of J_j scheduled in the rth position. The actual resource allocated to job J_j is denoted as u_j. For the linear resource consumption model, the actual processing time of job J_j is determined by

$$p_{jr} = p_j r^{a_j} - c_j u_j, \tag{2}$$

where c_j is the positive compression rate of job J_j, $0 \le u_j \le \bar{u}_j$ and $p_{jr} \ge 0$.

The RMA duration is determined by $f(t) = b + \sigma t$, where $b > 0$ is the basic RMA time, $\sigma \ge 0$ is the RMA deterioration rate, and t is the starting time of the RMA. The modifying rate of job J_j is notated by λ_j. After the RMA, the machine will revert to its initial conditions, machine deterioration will start anew, and the processing time of job J_j will be multiplied by λ_j. In the proposed model, we assume the RMA can improve the efficiency of a job (therefore λ_j takes a value in (0, 1]).

The due-window of job J_j is specified by a pair of non-negative real numbers $[d_j, d'_j]$ such that $d_j \le d'_j$, where d_j and d'_j are the beginning and ending times of the due-window respectively. For the slack due-window method, d_j and d'_j are determined by

$$d_j = p_{jr} + q, \tag{3}$$

and

$$d'_j = p_{jr} + q', \tag{4}$$

where $q' > q$ are two job-independent constants.

Then the due-window size $D_j = d_j - d'_j = q' - q$, for $j = 1, \dots, n$, is identical for all the jobs. Let $D = D_j$.

For a given schedule π, C_j denotes the completion time of job J_j, $E_j = \max\{0, d_j - C_j\}$ is the earliness value of job J_j, and $T_j = \max\{0, C_j - d'_j\}$ is the tardiness value of job J_j. To this end, we can create the following total cost function

$$Z = \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \gamma d_j + \delta D) + \theta \sum_{j=1}^{n} G_j u_j, \tag{5}$$

which takes into account (i) earliness E_j, (ii) tardiness T_j, (iii) the starting time of the due-window d_j, (iv) the due-window size D, and (v) the resource allocation. We further define $\alpha > 0$, $\beta > 0$, $\gamma > 0$ and $\delta > 0$ representing the earliness, tardiness, due-window starting time and due-window size costs per unit time respectively. For the resource consumption cost, G_j is defined as the per unit resource cost for job J_j and θ is a constant weight which is specified by the decision-maker.

The general objective is to determine the optimal job sequence, the optimal location of the RMA, the optimal resource consumption and the optimal q and q' to minimize the total cost function

$$Z = Z(\pi, u, q, q') = \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \gamma d_j + \delta D) + \theta \sum_{j=1}^{n} G_j u_j. \tag{6}$$

The problems under study are denoted as

$$1 \mid SLK, p_{jr} = p_j r^{a_j} - c_j u_j,$$

$$RMA \mid \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \gamma d_j + \delta D) + \theta \sum_{j=1}^{n} G_j u_j,$$

where SLK and RMA denote the slack due-window method and rate-modifying activity, respectively.

3. Optimal Solution

In this section some properties for an optimal schedule are obtained.

The proofs of the following lemmas are similar to those in [14] and [15]. We use a conventional notation $[r]$ to indicate the index of a job which is allocated at the rth position.

Lemma 1. *If $C_{[r]} \geq d'_{[r]}$ holds, then $C_{[r+1]} \geq d'_{[r+1]}$.*

Lemma 2. *If $C_{[r]} \leq d_{[r]}$ holds, then $C_{[r-1]} \leq d_{[r-1]}$.*

Consider a job sequence π and a resource consumption way $u = (u_1, u_2, \dots, u_n)$. Assume that $C_{[s]} \leq q \leq C_{[s+1]}$ and $C_{[t]} \leq q' \leq C_{[t+1]}$. Then the total cost Z is a linear function of q and q', and thus an optimum is obtained either at $q = C_{[s]}$ or $q = C_{[s+1]}$ and either at $q' = C_{[t]}$ or $q' = C_{[t+1]}$.

Lemma 3. *(i) For any given job sequence π and resource consumption u, there exists an optimal schedule in which the values of q and q' coincide with the completion times of the k-th and l-th jobs ($l \geq k$) in the sequence.*

(ii) An optimal schedule starts at time zero and contains no idle time between consecutive jobs.

For a number a, *thesymbol* $\lfloor a \rfloor$ denotes the largest integer not more than a.

Lemma 4. $k = \left\lfloor \frac{n(\delta-\gamma)}{\alpha} \right\rfloor$ *and* $l = \left\lfloor \frac{n(\beta-\delta)}{\beta} \right\rfloor$.

By Lemma 4, the values of k and l can be calculated. Let i be the position of the last job preceding the RMA. If the position of the RMA is before k (i.e., $i < k$), then the total cost is given by

$$\begin{aligned}
Z &= \sum_{r=1}^{n}(\alpha E_{[r]} + \beta T_{[r]} + \gamma d_{[r]} + \delta D) + \theta \sum_{j=1}^{n} G_j u_j \\
&= \alpha \sum_{r=1}^{k}(p_{jr} + q - C_{[r]}) + \beta \sum_{r=l+1}^{n}(C_{[r]} - p_{jr} - q') \\
&+ \gamma \sum_{r=1}^{n}(q + p_{jr}) + n\delta(q' - q) + \theta \sum_{j=1}^{n} G_j u_j \\
&= \alpha \sum_{r=1}^{k} j p_{jr} + \alpha i(b + \sigma \sum_{r=1}^{i} p_{jr}) + \beta \sum_{r=l+1}^{n}(n-r)p_{jr} \\
&+ \gamma(n(b + \sigma \sum_{r=1}^{i} p_{jr}) + (n+1)\sum_{r=1}^{k} p_{jr} + \sum_{r=k+1}^{n} p_{jr})
\end{aligned}$$

$$+ n\delta \sum_{r=k+1}^{l} p_{jr} + \theta \sum_{j=1}^{n} G_j u_j$$
$$= nb\gamma + \alpha ib + \sum_{r=1}^{n} w_r p_{jr} + \theta \sum_{j=1}^{n} G_j u_j, \tag{7}$$

where

$$w_r = \begin{cases} \alpha r + \alpha i\sigma + \gamma n\sigma + (n+1)\gamma, & \text{if } r = 1, 2, ..., i, \\ \alpha r + (n+1)\gamma, & \text{if } r = i+1, i+2, ..., k, \\ \gamma + n\delta, & \text{if } r = k+1, k+2, ..., l, \\ \beta(n-r) + \gamma, & \text{if } r = l+1, l+2, ..., n. \end{cases} \tag{8}$$

If $k \leq i < l$, then we have

$$Z = \alpha \sum_{r=1}^{k} r p_{jr} + \beta \sum_{r=l+1}^{n} (n-r) p_{jr} + \gamma \left((n+1) \sum_{r=1}^{k} p_{jr} + \sum_{r=k+1}^{n} p_{jr} \right)$$
$$+ n\delta(b + \sigma \sum_{r=1}^{i} p_{jr}) + n\delta \sum_{r=k+1}^{l} p_{jr} + \theta \sum_{j=1}^{n} G_j u_j \tag{9}$$
$$= n\delta b + \sum_{r=1}^{n} w_r p_{jr} + \theta \sum_{j=1}^{n} G_j u_j,$$

where

$$w_r = \begin{cases} \alpha r + \gamma(n+1) + n\delta\sigma, & 1 \leq r \leq k, \\ \gamma + n\delta\sigma + n\delta, & k < r \leq i, \\ \gamma + n\delta, & i < r \leq l, \\ \beta(n-r) + \gamma, & l < r \leq n. \end{cases} \tag{10}$$

If $l \leq i \leq n$, then we have

$$Z = \alpha \sum_{r=1}^{k} r p_{jr} + \beta \left(\sum_{r=l+1}^{n} (n-r) p_{jr} + (n-i)(b + \sigma \sum_{r=1}^{i} p_{jr}) \right)$$
$$+ \gamma \left((n+1) \sum_{r=1}^{k} p_{jr} + \sum_{r=k+1}^{n} p_{jr} \right) + n\delta \sum_{r=k+1}^{l} p_{jr} + \theta \sum_{j=1}^{n} G_j u_j \tag{11}$$
$$= (n-i)\beta b + \sum_{r=1}^{n} w_r p_{jr} + \theta \sum_{j=1}^{n} G_j u_j,$$

where

$$w_r = \begin{cases} \alpha r + \beta(n-i)\sigma + \gamma(n+1), & 1 \le r \le k, \\ \beta(n-i)\sigma + \gamma + n\delta, & k < r \le l, \\ \beta(n-r) + \beta(n-i)\sigma + \gamma, & l < r \le i, \\ \beta(n-r) + \gamma, & i < r \le n. \end{cases} \tag{12}$$

Note that when $i = n$ there is no RMA scheduled since by then all jobs are finished.

For the linear resource consumption models, we have

$$p_{jr} = \begin{cases} p_j r^{a_j} - c_j u_j, & r \le i, \\ \lambda_j p_j (r-i)^{a_j} - c_j u_j, & r > i, \end{cases} \tag{13}$$

and thus

$$Z = M + \sum_{r=1}^{n} w_r p_j \eta_{jr} + \sum_{r=1}^{n} (\theta G_j - w_r c_j) u_j, \tag{14}$$

where

$$M = \begin{cases} nb\gamma + \alpha ib, & i < k, \\ n\delta b, & k \le i < l, \\ (n-i)\beta b, & l \le i \le n, \end{cases} \tag{15}$$

and

$$\eta_{jr} = \begin{cases} r^{a_j}, & r \le i, \\ \lambda_j (r-i)^{a_j}, & r > i. \end{cases} \tag{16}$$

Therefore, minimizing the objective function Z is equivalent to minimizing Z':

$$Z' = \sum_{r=1}^{n} w_r p_j \eta_{jr} + \sum_{r=1}^{n} (\theta G_j - w_r c_j) u_j. \tag{17}$$

Since $p_{jr} > 0$, we have

$$p_j r^{a_j} - c_j u_j \ge 0 \ \text{ if } \ r \le i \tag{18}$$

and

$$\lambda_j p_j (r-i)^{a_j} - c_j u_j \ge 0 \ \text{ if } \ r > i. \tag{19}$$

Set

$$u_j' = \begin{cases} \min\{\overline{u}_j, \frac{p_j r^{a_j}}{c_j}\}, & \text{if } r \le i, \\ \min\{\overline{u}_j, \frac{\lambda_j p_j (r-i)^{a_j}}{c_j}\}, & \text{if } r > i. \end{cases} \tag{20}$$

Therefore, we have $u_j \leq u'_j$.

We see the optimal resource consumption for a job depends on the sign of $\theta G_j - w_r c_j$. Let u_j^* be the optimal resource consumption for job J_j. Then,

$$u_j^* = \begin{cases} u'_j, & \text{if } \theta G_j - w_r c_j < 0, \\ u_j \in [0, u'_j], & \text{if } \theta G_j - w_r c_j = 0, \\ 0, & \text{if } \theta G_j - w_r c_j > 0. \end{cases} \tag{21}$$

To this end we can define the element χ_{jr} in an assignment matrix as follows:

$$\chi_{jr} = \begin{cases} w_r p_j \eta_{jr}, & \text{if } \ \theta G_j - w_r c_j \geq 0, \\ w_r p_j \eta_{jr} + (\theta G_j - w_r c_j) u'_j, & \text{if } \ \theta G_j - w_r c_j < 0, \end{cases} \tag{22}$$

and

$$z_{jr} = \begin{cases} 1 & \text{if job } J_j \text{ is scheduled in the } r\text{th position} \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

To minimize the problem $1 \mid SLK, p_{jr} = p_j r^{a_j} - c_j u_j, RMA \mid \sum_{j=1}^{n} (\alpha E_j + \beta T_j + \gamma d_j + \delta D) + \theta \sum_{j=1}^{n} G_j u_j$ is equivalent to minimizing the following Assignment Problem, which can be solved in time complexity $O(n^3)$:

$$\min \sum_{j=1}^{n} \sum_{r=1}^{n} \chi_{jr} z_{jr}$$
$$\text{s.t.} \begin{cases} \sum_{j=1}^{n} z_{jr} = 1, & r = 1, 2, \ldots, n, \\ \sum_{r=1}^{n} z_{jr} = 1, & j = 1, 2, \ldots, n, \\ z_{jr} = 0 \text{ or } 1, & j, r = 1, 2, \ldots, n. \end{cases} \tag{24}$$

Here (24) presents a 0-1 integer linear programming problem, which guarantees that each position has one job scheduled and each job is scheduled once. The following solution algorithm has an $O(n^4)$ time complexity.

Algorithm 1. Solution algorithm for the problem $1 \mid SLK, p_{jr} = p_j r^{a_j} - c_j u_j, RMA \mid \sum_{j=1}^{n} (\alpha E_j + \beta T_j + \gamma d_j + \delta D) + \theta \sum_{j=1}^{n} G_j u_j$.

```
SET $k = \left\lfloor \frac{n(\delta-\gamma)}{\alpha} \right\rfloor$, $l = \left\lfloor \frac{n(\beta-\delta)}{\beta} \right\rfloor$.
FOR each position $i = 0, 1, \ldots, n$ available to allocate
rate-modifying activity
    FOR each position $r = 1, 2, \ldots, n$ in a schedule
```

```
4           DETERMINE the positional weight $w_r$
5       END FOR
6       FOR each job $j = 1, 2, \ldots, n$
7           FOR each position $r = 1, 2, \ldots, n$ in a schedule
8               DETERMINE the value $\chi_{jr}$ according to (22)
9           END FOR
10      END FOR
11      DETERMINE a local optimal schedule of the
        assignment problem described in (24) and its
        total cost
12 END FOR
13 DETERMINE the global optimal schedule with the
    minimum total cost
```

Theorem 1. *Algorithm 1 solves the problem* $1 \mid SLK, p_{jr} = p_j r^{a_j} - c_j u_j, RMA \mid \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \gamma d_j + \delta D) + \theta \sum_{j=1}^{n} G_j u_j$ *in* $O(n^4)$ *time.*

Proof. The correction of Algorithm 1 is guaranteed by Lemmas 4 and the derivation from (7) to (24). The time complexity from Step 3 to Step 11 is $O(n^3)$. In the outer loop from Step 2 to Step 12, position index i takes on integer values between 0 to n. Hence, the time complexity for solving the $1 \mid SLK, p_{jr} = p_j r^{a_j} - c_j u_j, RMA \mid \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \gamma d_j + \delta D) + \theta \sum_{j=1}^{n} G_j u_j$ problem is $O(n^4)$. □

In most studies of the linear resource model, an upper bound on $\bar{u}$ is stated as (1). In this paper we remove this constraint and consider a more general case, to which Algorithm 1 gives a solution.

4. Numerical Example

In this section, Algorithm 1 for the linear resource model is demonstrated by the following example.

Example 1. There are $n = 7$ jobs. The initial setup of all jobs is illustrated by Table 1.

The penalties for unit earliness, tardiness, due-window starting time and due-window size are $\alpha = 2$, $\beta = 18$, $\gamma = 4$ and $\delta = 5$, respectively. The basic maintenance time is $b = 15$ and the deteriorating maintenance factor is $\sigma = 0.1$. The constant weight for resource consumption $\theta = 0.8$.

j	1	2	3	4	5	6	7
p_j	45	18	48	46	50	43	21
a_j	0.3	0.5	0.25	0.3	0.15	0.4	0.2
c_j	3	2	3	4	3	4	2
λ_j	0.58	0.84	0.45	0.55	0.38	0.64	0.75
G_j	34	40	25	38	24	48	42
$\overline{u}_j$	99	7	7	6	6	6	7

Table 1: The settings in Example 1.

Solution: By Lemma 4, we have the locations of $k = \left\lfloor \frac{n(\delta-\gamma)}{\alpha} \right\rfloor = 3$ and $l = \left\lfloor \frac{n(\beta-\delta)}{\beta} \right\rfloor = 5$.

As shown in Table 2, all the local optimal job sequences and the corresponding total costs are presented, among which the optimal total cost is underlined. The global optimal solution for this example includes the following: (i) the job sequence is (2, 4, 3, 5, 1, 7, 6) and the corresponding job starting time and actual processing time are (0, 19.40, 20.70, 25.39, 29.79, 29.79, 37.52) and (4.00, 1.30, 4.69, 4.40, 0, 7.73, 56.35), respectively; (ii) the slack window parameters are $q = 25.39$ and $q' = 29.79$; (iii) the RMA is located immediately after the first job (i.e. Job J_2), starting at time $t = 4.00$ and ending at time $t = 19.40$ (maintenance duration 15.40); (iv) the optimal resource consumption of each job is (13.19, 7.00, 7.00, 6.00, 6.00, 0, 7.00); (v) the total cost is $Z = 2645.61$.

i	Job sequence	Z
0	(2, 4, 3, 5, 1, 7, 6)	2915.80
1	(2, 4, 3, 5, 1, 7, 6)	$\underline{2645.61}$
2	(2, 1, 4, 3, 5, 7, 6)	2790.61
3	(2, 7, 1, 4, 3, 5, 6)	3060.96
4	(6, 2, 7, 1, 3, 5, 4)	3989.78
5	(6, 4, 2, 7, 1, 3, 5)	5616.00
6	(6, 4, 2, 7, 1, 5, 3)	6290.07
7	(6, 4, 2, 7, 1, 5, 3)	6012.09

Table 2: The corresponding local optimal job sequences and total costs with one RMA at all possible positions in Example 1.

5. Conclusion

We have investigated a single machine scheduling and slack due-window assignment problem with linear resource allocation, aging effect and a deteriorating rate-modifying maintenance activity. The objective is to minimize the total cost caused by the due-window location, due-window size, earliness, tardiness and resource consumption.

Further research may consider the problem with other performance measures, or the problem with multiple rate-modifying maintenance activities.

References

[1] T. C. E. Cheng, Optimal common due-date with limited completion time deviation, *Computers & Operations Research*, **15** (1988), 91–96.

[2] S. Liman, S. Panwalkar, and S. Thongmee, Common due window size and location determination in a single machine scheduling problem, *Journal of the Operational Research Society*, **49** (1998), 1007–1010.

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

[4] 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*, **37** (2013), 8394–8400.

[5] S.-J. Yang, D.-L. Yang, and T. C. E. Cheng, Single-machine due-window assignment and scheduling with job-dependent aging effects and deteriorating maintenance, *Computers & Operations Research*, **37** (2010), 1510–1514.

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

[7] R. Rudek, Scheduling problems with position dependent job processing times: computational complexity results, *Annals of Operations Research*, **196** (2012), 491–516.

[8] R. Vickson, Two single machine sequencing problems involving controllable job processing times, *AIIE Transactions*, **12** (1980), 258–262.

[9] M. Ji, J. Ge, K. Chen, and T. Cheng, Single-machine due-window assignment and scheduling with resource allocation, aging effect, and a deteriorating rate-modifying activity, *Computers & Industrial Engineering*, **66** (2013), 952–961.

[10] B. Cheng and L. Cheng, Note on single-machine due-window assignment and scheduling with resource allocation, aging effect, and a deteriorating rate-modifying activity, *Computers & Industrial Engineering*, **78** (2014), 320–322.

[11] C.-Y. Lee and V. Leon, Machine scheduling with a rate-modifying activity, *European Journal of Operational Research*, **128** (2001), 119–128.

[12] T. C. E. Cheng, S.-J. Yang, and D.-L. Yang, Common due-window assignment and scheduling of linear time-dependent deteriorating jobs and a deteriorating maintenance activity, *International Journal of Production Economics*, **135** (2012), 154–161.

[13] B. Mor and G. Mosheiov, Scheduling a maintenance activity and due-window assignment based on common flow allowance, *International Journal of Production Economics*, **135** (2012), 222–230.

[14] G. Mosheiov and D. Oron, Job-dependent due-window assignment based on a common flow allowance, *Foundations of Computing and Decision Sciences*, **35** (2010), 185–195.

[15] K. Chen, M. Ji, and J. Ge, A note on scheduling a maintenance activity and due-window assignment based on common flow allowance, *International Journal of Production Economics*, **145** (2013), 645–646.