International Journal of Applied Mathematics

Volume 31 No. 5 2018, 637-647
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
doi: http://dx.doi.org/10.12732/ijam.v31i5.9

SINGLE MACHINE SLACK DUE-WINDOW SCHEDULING WITH LINEAR RESOURCE ALLOCATION AND POSITION-DEPENDENT PROCESSING TIMES

Bo Cheng[1 §], Ling Cheng[2]

[1]Department of Applied Mathematics
School of Finance
Guangdong University of Foreign Studies
Guangzhou, 510420, 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 investigate single machine scheduling with linear resource allocation and position-dependent processing times based on the slack due-window method. The objective is to minimize the total cost caused by the due-window location, the due-window size, the resource consumption, the makespan and the earliness and tardiness with respect to a slack due-window. We provide a polynomial-time algorithm to solve the problem.

AMS Subject Classification: 90B35
Key Words: resource allocation, due-window, machine scheduling, resource consumption model

1. Introduction

In the classical scheduling theory, job processing times are considered as constants. However, in the last decade we have witnessed a steadily growing interest on solving scheduling problems with changeable job processing times. There is a practical merit of considering changeable job processing times. In practice, the processing time of a job can be dependent on the position scheduled, and

Received: June 25, 2018

§Correspondence author

the phenomenon that the actual processing time of a job can be reduced due to an additional resource allocated to the job has been noted. It prompts the studies on a variety of single machine scheduling models and assignment problems taking into account the effects of resource allocation and position-dependent processing times.

Conventionally, there are a few due-window methods, for example, common due-window and slack due-window. In this paper, we investigate the single machine scheduling and slack due-window assignment problem with resource allocation and position-dependent processing times. To our best knowledge, this problem has not been studied in literatures. In a relevant work, [1] investigated a similar model but with position-independent processing times. In this paper, we generalize this model by assuming that the processing times are position-dependent.

The rest of this paper is organized as follows. The problem under study is illustrated in Section 2. The optimal (polynomial-time) solution is presented in Section 3. A numerical example to demonstrate the polynomial-time solution is given in Section 4. The research is concluded and future study is foreseen in the last section.

2. Model Formulation

In the models of this paper, n independent and non-preemptive jobs $J_1, J_2, \ldots, J_n$ are scheduled on a single machine. All the jobs can be arranged at time zero. Let a_{jr} and p_{jr} be the normal and the actual position-dependent processing times of job J_j when it is arranged in the rth position. The actual and the maximum-available resource allocated to job J_j are denoted by u_j and $\bar{u}_j$, respectively. For the linear resource consumption model, the actual processing time of job J_j is determined by

$$p_{jr} = a_{jr} - c_j u_j, \tag{1}$$

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

The due-window of job J_j is determined by a pair of non-negative real numbers $[d_j, d'_j]$ such that $d_j \leq d'_j$, 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 decided by

$$d_j = p_{jr} + q, \tag{2}$$

and

$$d'_j = p_{jr} + q', \tag{3}$$

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

Then the due-window size $D_j = d_j - d'_j = q' - q$, for $j = 1, \ldots, n$, is equal 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. The makespan $C_{max} = \max\{C_j | j = 1, 2, \ldots, n\}$ is the completion time of all jobs. Here C_{max} is the completion time of the last job scheduled in the n'th position.

To this end, we can create the following total cost function

$$Z = \sum_{t=1}^{n} (\alpha E_t + \beta T_t + \gamma d_t + \delta D) + \eta \sum_{t=1}^{n} G_t u_t + \theta C_{max}, \tag{4}$$

which takes into account (i) earliness E_j, (ii) tardiness T_j, (iii) starting time of the due-window d_j, (iv) due-window size D, (v) resource consumption u_j and (vi) makespan C_{max}. 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_t is defined as the per unit resource cost for job J_t. Here $\eta \geq 0$ and $\theta \geq 0$ are two constant weights which are specified by the decision-maker.

With the three-field notation of [2], the problem under study is expressed as $1 \mid SLKW, p_{jr} = a_{jr} - c_j u_j \mid \sum_{t=1}^{n} (\alpha E_t + \beta T_t + \gamma d_t + \delta D) + \eta \sum_{t=1}^{n} G_t u_t + \theta C_{max}$, where $SLKW$ in the second field denotes the slack due-window method.

3. Optimal Solution Algorithm

In this section we present some properties for a schedule.The proofs of the following lemmas are similar to those in [3], [4], [5] and [6]. 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 allocation way $u = (u_1, u_2, \ldots, u_n)$. Suppose $C_{[s]} \leq q \leq C_{[s+1]}$ and $C_{[t]} \leq q' \leq C_{[t+1]}$. Then the total cost Z is a linear function of q and q', and hence 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 resource allocation u and job sequence π, there exists an optimal scheduling such that 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 scheduling begins at time zero and has no idle time between consecutive jobs.

For a number a, $\lfloor a \rfloor$ denotes the largest integer less than or equal to a.

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

With Lemma 4, the values of k and l can be obtained. In the following we assume $k \leq l$ and refer to [7] for the other cases.

For the objective function, we have

$$\begin{aligned} Z &= \sum_{r=1}^{n}(\alpha E_{[r]} + \beta T_{[r]} + \gamma d_{[r]} + \delta D) + \eta \sum_{t=1}^{n} G_t u_t + \theta C_{max} \\ &= \sum_{r=1}^{n} w_r p_{jr} + \eta \sum_{t=1}^{n} G_t u_t, \end{aligned} \tag{5}$$

where

$$w_r = \begin{cases} \alpha r + \gamma(n+1) + \theta, & 1 \leq r \leq k, \\ \gamma + n\delta + \theta, & k < r \leq l, \\ \beta(n-r) + \gamma + \theta & l < r \leq n. \end{cases} \tag{6}$$

Since $p_{jr} = a_{jr} - c_j u_j$, we have

$$Z = \sum_{r=1}^{n} w_r a_{jr} + \sum_{r=1}^{n} (\eta G_j - w_r c_j) u_j, \tag{7}$$

where $j = [r]$.

Due to the fact that $p_{jr} \geq 0$, we have $u_j \leq \frac{a_{jr}}{c_j}$. Set $u'_j = \min\{\overline{u}_j, \frac{a_{jr}}{c_j}\}$, and hence we have $u_j \leq u'_j$.

Note that the optimal resource allocation for job J_j depends on the sign of $\eta G_j - w_r c_j$. Let u^*_j be the optimal resource allocation for job J_j. Then

$$u^*_j = \begin{cases} u'_j, & \text{if } \eta G_j - w_r c_j < 0, \\ u_j \in [0, u'_j], & \text{if } \eta G_j - w_r c_j = 0, \\ 0, & \text{if } \eta G_j - w_r c_j > 0. \end{cases} \tag{8}$$

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

$$\chi_{jr} = \begin{cases} w_r a_{jr}, & \text{if } \eta G_j - w_r c_j \geq 0, \\ w_r a_{jr} + (\eta G_j - w_r c_j) u'_j, & \text{if } \eta G_j - w_r c_j < 0, \end{cases} \tag{9}$$

and

$$z_{jr} = \begin{cases} 1 & \text{if job } J_j \text{ is arranged in the } r\text{th position} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

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

$$\begin{gathered} \min \sum_{j=1}^{n} \sum_{r=1}^{n} \chi_{jr} z_{jr} \\ \text{s.t.} \begin{cases} \sum_{j=1}^{n} z_{jr} = 1, & r = 1, 2, \ldots, n, \\ \sum_{r=1}^{n} z_{jr} = 1, & j = 1, 2, \ldots, n, \\ z_{jr} = 0 \text{ or } 1, & j, r = 1, 2, \ldots, n. \end{cases} \end{gathered} \tag{11}$$

Algorithm 1. Solution algorithm for the problem $1 \mid SLKW$, $p_{jr}{=}a_{jr} - c_j u_j \mid \sum_{t=1}^{n}(\alpha E_t + \beta T_t + \gamma d_t + \delta D) + \eta \sum_{t=1}^{n} G_t u_t + \theta C_{max}$.

1 SET $k = \left\lfloor \frac{n(\delta-\gamma)}{\alpha} \right\rfloor$, $l = \left\lfloor \frac{n(\beta-\delta)}{\beta} \right\rfloor$.
2 FOR each position $r = 1, 2, \ldots, n$ in a schedule
3 DETERMINE the positional weight w_r
4 END FOR
5 FOR each job $j = 1, 2, \ldots, n$
6 FOR each position $r = 1, 2, \ldots, n$ in a schedule
7 DETERMINE the value χ_{jr} according to (9)
8 END FOR
9 END FOR
10 DETERMINE the global optimal schedule of the assignment problem described in (11) and its total cost

Theorem 1. *Algorithm 1 solves the problem* $1 \mid SLKW, p_{jr} = a_{jr} - c_j u_j \mid \sum_{t=1}^{n}(\alpha E_t + \beta T_t + \gamma d_t + \delta D) + \eta \sum_{t=1}^{n} G_t u_t + \theta C_{max}$ *in* $O(n^3)$ *time.*

Proof. The correction of Algorithm 1 is guaranteed by Lemmas 4 and the deduction from (5) to (11). The time complexity from Step 2 to Step 4 is $O(n)$, the time complexity from Step 5 to Step 9 is $O(n^2)$, and the time complexity of Step 10 is $O(n^3)$. Hence, the time complexity for solving the $1 \mid SLKW, p_{jr} = a_{jr} - c_j u_j \mid \sum_{t=1}^{n}(\alpha E_t + \beta T_t + \gamma d_t + \delta D) + \eta \sum_{t=1}^{n} G_t u_t + \theta C_{max}$ problem is $O(n^3)$. $\square$

Remark. Theorem 3.3 in [1] can be regarded as a special case of Theorem 1.

4. Numerical Example

In this section, Algorithm 1 for the linear resource model presented in Section 3 is demonstrated by the following example.

Example 1. There are $n = 7$ jobs. The initial settings of all jobs are illustrated by Table 1 and Table 2.

j	1	2	3	4	5	6	7
c_j	3	2	3	4	3	4	2
G_j	34	40	25	38	24	48	42
$\overline{u}$	8	7	7	6	6	6	7

Table 1: The settings in Example 1.

a_{jr}	$r = 1$	2	3	4	5	6	7
$j = 1$	50	30	35	40	55	60	70
2	40	20	45	50	40	70	30
3	35	40	55	50	30	45	22
4	45	52	63	28	41	27	39
5	53	62	74	35	27	39	40
6	30	20	54	62	39	40	50
7	30	50	20	30	40	50	60

Table 2: The a_{jr} settings in Example 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 constant weights are specified by the decision-maker as $\theta = 0.8$ and $\eta = 1$.

Solution. By Lemma 4, we have the locations of $k = \left\lfloor \frac{n(\delta-\gamma)}{\alpha} \right\rfloor = 3$ and $l = \left\lfloor \frac{n(\beta-\delta)}{\beta} \right\rfloor = 5$. First, we obtain positional weights w_r for $r = 1, 2, \ldots, 7$ as shown in Table 3.

r	1	2	3	4	5	6	7
w_r	34.8	36.8	38.8	39.8	39.8	22.8	4.8
Job sequence	6	1	7	4	3	5	2

Table 3: Positional weights in Example 1.

Next, based on (9), we obtain χ_{jr} in the following assignment matrix:

$$\begin{bmatrix} 1176.80 & 492.80 & 698.80 & 908.80 & 1505.80 & 1092.80 & 336.00 \\ 1184.80 & 500.80 & 1482.80 & 1712.80 & 1314.80 & 1556.80 & 144.00 \\ 662.20 & 874.20 & 1494.20 & 1329.20 & 533.20 & 722.20 & 105.60 \\ 958.80 & 1258.40 & 1741.20 & 387.20 & 904.60 & 296.40 & 187.20 \\ 1362.00 & 1763.20 & 2316.80 & 820.60 & 502.20 & 622.80 & 192.00 \\ 496.80 & 240.00 & 1452.00 & 1800.40 & 885.00 & 652.80 & 240.00 \\ 850.80 & 1618.80 & 526.80 & 930.80 & 1328.80 & 1114.80 & 288.00 \end{bmatrix} \tag{12}$$

By solving the assignment problem described in (11), we obtain the local optimal sequence (6, 1, 7, 4, 3, 5, 2) and the total cost $Z = 3203.60$.

The global optimal solution for this example includes the following: (i) the job sequence is (6, 1, 7, 4, 3, 5, 2) and the corresponding job starting time and actual processing time are (0, 6, 12, 18, 22, 31, 52) and (6, 6, 6, 4, 9, 21, 30), respectively; (ii) the slack window parameters are $q = 18$ and $q' = 31$; (iii) the optimal resource consumption of each job is (8, 0, 7, 6, 6, 6, 7); (iv) the total cost is $Z = 3203.60$.

5. Conclusion

We solved the single machine scheduling and slack due-window assignment problems with position-dependent processing times for the linear resource allocation models, and presented a polynomial-time solutions to minimize the total cost caused by the due-window location, due-window size, earliness, tardiness, resource consumption and makespan. Further research may consider the problem with other objective functions, or the problem with other settings, e.g., production and delivery batch scheduling [8], [9] and [10].

References

[1] 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*, **65**, No 1 (2014), 1–13.

[2] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. Kan, Optimization and approximation in deterministic sequencing and scheduling: A survey, *Annals of Discrete Mathematics*, **5** (1979), 287–326.

[3] 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.

[4] B. Mor and G. Mosheiov, Scheduling a maintenance activity and due-window assignment based on common flow allowance, *International Journal of Production Economics*, **135**, No 1 (2012), 222–230.

[5] 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**, No 2 (2013), 645–646.

[6] B. Cheng and L. Cheng, Single machine slack due-window scheduling with linear resource allocation, aging effect, and a deteriorating rate-modifying activity, *International Journal of Applied Mathematics*, **30**, No 5 (2017), 375–386; DOI: 10.12732/ijam.v30i5.2.

[7] 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**, No 4 (2013), 952–961.

[8] E. Prasetyaningsih, T. Samadhi, and A. Halim, Production and delivery batch scheduling with a common due date and multiple vehicles to minimize total cost, *IOP Conference Series: Materials Science and Engineering*, **114**, No 1 (2016), 1–10.

[9] E. Prasetyaningsih, Suprayogi, T. M. A. A. Samadhi, and A. H. Halim, Production and delivery batch scheduling with multiple due dates to minimize total cost, *Journal of Engineering and Technological Science*, **49**, No 1 (2017), 16–36.

[10] N. P. A. Hidayat, A. Cakravastia, T. M. A. A. Samadhi, and A. H. Halim, A batch scheduling for m heterogeneous batch processor, *International Journal of Production Research*, **54**, No 4 (2016), 1170–1185.