**Open Access Library Journal**
2018, Volume 5, e4907
ISSN Online: 2333-9721
ISSN Print: 2333-9705

# Single Machine Slack Due-Window Assignment and Scheduling of Linear Time-Dependent Deteriorating Jobs and a Deteriorating Maintenance Activity

**Bo Cheng[1], Ling Cheng[2]**

[1]Department of Applied Mathematics, School of Finance, Guangdong University of Foreign Studies, Guangzhou, China
[2]School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg, South Africa
Email: 200511359@oamail.gdufs.edu.cn, ling.cheng@wits.ac.za

**How to cite this paper:** Cheng, B. and Cheng, L. (2018) Single Machine Slack Due-Window Assignment and Scheduling of Linear Time-Dependent Deteriorating Jobs and a Deteriorating Maintenance Activity. *Open Access Library Journal*, **5**: e4907.
https://doi.org/10.4236/oalib.1104907

**Received:** September 30, 2018
**Accepted:** October 8, 2018
**Published:** October 11, 2018



## Abstract

In this paper, we consider the slack due-window assignment model and study a single machine scheduling problem of linear time-dependent deteriorating jobs and a deteriorating maintenance activity. The objective is to find the job schedule having an assigned maintenance activity and due-windows with the minimum total cost consisting of costs of earliness, tardiness, window location and window size. A polynomial-time algorithm is presented in this paper with time complexity $O(n^2 \log n)$ for $n$ jobs.

## Subject Areas

Operational Research

## Keywords

Deteriorating Job, Due-Window, Maintenance Activity, Scheduling

## 1. Introduction

Competition in market place prompts the studies on operations management to improve customer service. One important objective of operations management in practice is to finish jobs as close as possible to their due-dates. Usually a time period, namely due-window of a job, is assigned in the supply contract so that a job completed within the time period will not be penalized. The due-window assignment methods include common due-window, slack due-window (also called common flow allowance) and others. Some relevant references are [1] [2] etc. A polynomial-time solution was introduced to obtain the optimal schedule and the optimal due-window size with the minimum total cost in [3].

In classical scheduling theory, job processing times are considered as constants. However, a steadily growing interest on solving scheduling problems with changeable job processing times has been witnessed in the last decade. Kang *et al.* [4] showed that the problem of minimizing makespan on $m$ identical parallel machines with time-dependent processing times is NP-hard, and presented a fully polynomial-time approximation scheme for the problem. The single machine common due-window assignment problem considering deteriorating jobs and learning effect was studied by [5]. A polynomial-time algorithm was presented to give a solution to minimize the total cost consisting of the penalties of earliness, tardiness, window location and window size. Yin *et al.* [6] investigated the single machine scheduling problems with sum-of-logarithm-processing-times based deterioration.

Machine scheduling for a rate-modifying activity was first studied by [7]. In this paper, the maintenance activity considered is different from the rate-modifying activity. At most one maintenance activity is scheduled and the scheduler can decide when to start the maintenance activity. After the maintenance activity the machine reverts to its initial conditions including machine deterioration. The research on the similar assumption can be found in [8] and [9].

The combinations of the above-mentioned settings have been considered in the following recent literatures. Based on the common due-window assignment method, Zhao and Tang [10] studied the scheduling problem with a rate-modifying activity and time-dependent deteriorating jobs, and Cheng *et al.* [9] investigated the problem with a maintenance activity and time-dependent deteriorating jobs. [11] and [12] studied the problem to minimize the total cost consisting of earliness, tardiness, due-window starting time, due-window size, and resource consumption with a common due-window and a deteriorating rate-modifying activity. In this paper, the problem of slack due-window assignment and single-machine scheduling considering a maintenance activity and time-dependent deteriorating jobs is presented. To our best knowledge, this problem has not been studied in literatures.

The rest of this paper is organized as follows. In Section 2, a description of the problem is given. In Section 3, some important lemmas and properties are presented. In Section 4, a polynomial-time solution for the problem is given. A numerical example is presented to demonstrate the polynomial-time solution in Section 5. The research is concluded and future study is foreseen in the last section.

## 2. Model Formulation

A single machine processes $n$ jobs denoted by $J_1, J_2, \cdots, J_n$. Here any job is ready for processing at time zero. No preemption is allowed. Let $a_j$ denote the normal processing time of job $J_j$ and let non-negative *t* denote the starting time of job $J_j$. A common deteriorating factor *b* is specified for all the jobs. The actual processing time $p_j$ of job $J_j$ is determined by $p_j = a_j + bt$ according to a linear time-dependent deteriorating model.

The due window of job $J_j$ is specified by a pair of non-negative real numbers $\left[d_j^{(1)}, d_j^{(2)}\right]$ such that $d_j^{(1)} \le d_j^{(2)}$. For a given schedule $\pi$, $C_j = C_j(\pi)$ denotes the completion time of job $J_j$, $E_j = \max\left\{0, d_j^{(1)} - C_j\right\}$ is the earliness value of job $J_j$, $T_j = \max\left\{0, C_j - d_j^{(2)}\right\}$ is the tardiness value of job $J_j$, and $D_j = d_j^{(2)} - d_j^{(1)}$ is the due-window size of job $J_j$. For the slack due-window method, the due window starting time $d_j^{(1)}$ and the due window completion time $d_j^{(2)}$ for job $J_j$ are defined as

$$d_j^{(1)} = p_j + q^{(1)}, \tag{1}$$

and

$$d_j^{(2)} = p_j + q^{(2)}, \tag{2}$$

respectively. Note that $p_j$ is the processing time of job $J_j$. Since $q^{(1)}$ and $q^{(2)}$ are two job-independent constants, which satisfy $q^{(2)} > q^{(1)}$, the window size $D_j = q^{(2)} - q^{(1)}$ is also a constant for all the jobs. Let $D = D_j$.

Furthermore, the following assumptions have been made for this problem. First, the machine reverts to its initial conditions including machine deterioration after the maintenance activity. Second, there is at most one maintenance activity throughout the schedule. Note that it is unnecessary to allocate a maintenance activity after the final job. Third, a similar linear time-dependent deteriorating model is adopted to calculate maintenance duration. The duration $T_{ma}$ is determined by basic maintenance time $\mu$ (a positive constant), maintenance factor $\sigma$ and the starting time *t* of maintenance activity such as $T_{ma} = \mu + \sigma t$.

The objective function consists of four cost components, *i.e.* 1) earliness $E_j$, 2) tardiness $T_j$, 3) the starting time of the due-window $d_j^{(1)}$, and 4) the due-window size *D*. Let $\alpha > 0$, $\beta > 0$, $\gamma > 0$ and $\delta > 0$ represent the earliness, tardiness, due-window starting time and due-window size costs per unit time respectively. The general objective is to determine the optimal $q^{(1)}$ and $q^{(2)}$, the optimal position of the maintenance activity, and the optimal schedule to minimize the total cost function

$$Z = \sum_{j=1}^{n} \left(\alpha E_j + \beta T_j + \gamma d_j^{(1)} + \delta D\right). \tag{3}$$

The problem under study is denoted as

$$1 \mid SLK, p_j = a_j + bt, ma \mid \sum\nolimits_{j=1}^{n} \left(\alpha E_j + \beta T_j + \gamma d_j^{(1)} + \delta D\right)$$

where *SLK* and *ma* in the second field denote the slack due-window method and maintenance activity, respectively.

## 3. Properties of an Optimal Solution

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

**Lemma 1.** *If* $C_j \ge d_j^{(2)}$ *for a given job order* $\pi = (J_1, J_2, \cdots, J_n)$, *then* $C_{j+1} \ge d_{j+1}^{(2)}$.

**Proof:** We have

$$C_{j+1} \ge C_j + p_{j+1} \ge d_j^{(2)} + p_{j+1} = q^{(2)} + p_j + p_{j+1} = d_{j+1}^{(2)} + p_j \ge d_{j+1}^{(2)}.$$

□

Similar to the proof of Lemma 1, we have

**Lemma 2.** *If* $C_j \le d_j^{(1)}$ *for a given job order* $\pi = (J_1, J_2, \cdots, J_n)$, *then* $C_{j-1} \le d_{j-1}^{(1)}$.

Consider a job sequence $\pi = (J_1, J_2, \cdots, J_n)$. Assume that $C_s \le q^{(1)} \le C_{s+1}$ and $C_t \le q^{(2)} \le C_{t+1}$. Then the total cost $Z$ is a linear function of $q^{(1)}$ and $q^{(2)}$, and thus an optimum is obtained either at $q^{(1)} = C_s$ or $q^{(1)} = C_{s+1}$ and either at $q^{(2)} = C_t$ or $q^{(2)} = C_{t+1}$.

Therefore we obtain the following result, whose proof is similar to the one in [13].

**Lemma 3.** 1) For any given job sequence, the optimal values of $q^{(1)}$ and $q^{(2)}$ are equal to the completion times of the $k$th and $l$th jobs where $k \le l$.

2) An optimal schedule starts at time zero.

3) No idle time exists between consecutive jobs in an optimal schedule.

4) An optimal schedule exists in which the maintenance activity takes place right after the completion of one of the jobs.

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

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

**Proof:** Shift $q^{(1)}$ to the left by $\Delta$ time units, where $0 < \Delta < p_k$. As a result, the overall cost $Z$ has been changed by $(\delta n - n\gamma - \alpha k)\Delta$. Since $q^{(1)} = p_1 + p_2 + \cdots + p_k$ is optimal, it implies that $(\delta n - n\gamma - \alpha k)\Delta \ge 0$, and hence $k \le \dfrac{n(\delta - \gamma)}{\alpha}$.

Shift $q^{(1)}$ to the right by $\Delta$ time units, where $0 < \Delta < p_{k+1}$. As a result, the overall cost $Z$ has been changed by $(\alpha(k+1) + n\gamma - \delta n)\Delta$. We obtain $(\alpha(k+1) + n\gamma - \delta n)\Delta \ge 0$, and hence $k \ge \dfrac{n(\delta - \gamma)}{\alpha} - 1$. Then $k = \left\lfloor \dfrac{n(\delta - \gamma)}{\alpha} \right\rfloor$.

In the similar way, we can prove $l = \left\lfloor \dfrac{n(\beta - \delta)}{\beta} \right\rfloor$ by using the standard perturbation method. □

**Lemma 5.** Suppose that sequences $x_1, x_2, \cdots, x_n$ and $y_1, y_2, \cdots, y_n$ are given except in arrangement. The sum of the products of the corresponding elements $\sum_{j=1}^{n} x_j y_j$ is minimized if the sequences are monotonic in opposite senses.

**Proof:** See page 261 in [14]. □

## 4. Optimal Solution

Let $p_{[r]}$ denote the actual processing time and let $a_{[r]}$ denote the normal processing time of the job scheduled in the $r$th position in a sequence. The positions of $k$ and $l$ can be determined based on Lemma 4. Let $i$ be the position of the last job preceding the maintenance activity. If the position of the maintenance activity is before $k$ (*i.e.*, $i<k$), then the total cost is given by

$$\begin{aligned} Z &= \sum_{j=1}^{n}\left(\alpha E_j+\beta T_j+\gamma d_j^{(1)}+\delta D\right) \\ &= \alpha\sum_{j=1}^{k}\left(p_{[j]}+q^{(1)}-C_{[j]}\right)+\beta\sum_{j=l+1}^{n}\left(C_{[j]}-p_{[j]}-q^{(2)}\right) \\ &\quad +\gamma\sum_{j=1}^{n}\left(q^{(1)}+p_{[j]}\right)+n\delta\left(q^{(2)}-q^{(1)}\right) \\ &= \alpha\sum_{j=1}^{k} jp_{[j]}+\alpha i\left(\mu+\sigma\sum_{j=1}^{i}p_{[j]}\right)+\beta\sum_{j=l+1}^{n}(n-j)p_{[j]} \\ &\quad +\gamma\left(n\left(\mu+\sigma\sum_{j=1}^{i}p_{[j]}\right)+(n+1)\sum_{j=1}^{k}p_{[j]}+\sum_{j=k+1}^{n}p_{[j]}\right)+n\delta\sum_{j=k+1}^{l}p_{[j]} \\ &= n\mu\gamma+\alpha i\mu+\sum_{j=1}^{n}\omega_j p_{[j]}, \end{aligned} \tag{4}$$

where

$$\omega_j=\begin{cases}\alpha j+\alpha i\sigma+\gamma n\sigma+(n+1)\gamma & 1\le j\le i\\ \alpha j+(n+1)\gamma & i<j\le k\\ \gamma+n\delta & k<j\le l\\ \beta(n-j)+\gamma & l<j\le n\end{cases} \tag{5}$$

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

$$\begin{aligned} Z &= \alpha\sum_{j=1}^{k} jp_{[j]}+\beta\sum_{j=l+1}^{n}(n-j)p_{[j]}+\gamma\left((n+1)\sum_{j=1}^{k}p_{[j]}+\sum_{j=k+1}^{n}p_{[j]}\right) \\ &\quad +n\delta\left(\mu+\sigma\sum_{j=1}^{i}p_{[j]}\right)+n\delta\sum_{j=k+1}^{l}p_{[j]} \\ &= n\delta\mu+\sum_{j=1}^{n}\omega_j p_{[j]}, \end{aligned} \tag{6}$$

where

$$\omega_j=\begin{cases}\alpha j+\gamma(n+1)+n\delta\sigma & 1\le j\le k\\ \gamma+n\delta\sigma+n\delta & k<j\le i\\ \gamma+n\delta & i<j\le l\\ \beta(n-j)+\gamma & l<j\le n\end{cases} \tag{7}$$

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

$$
\begin{aligned}
Z &= \alpha \sum_{j=1}^{k} j p_{[j]} + \beta \left( \sum_{j=l+1}^{n} (n-j) p_{[j]} + (n-i) \left( \mu + \sigma \sum_{j=1}^{i} p_{[j]} \right) \right) \\
&\quad + \gamma \left( (n+1) \sum_{j=1}^{k} p_{[j]} + \sum_{j=k+1}^{n} p_{[j]} \right) + n\delta \sum_{j=k+1}^{l} p_{[j]} \\
&= (n-i) \beta\mu + \sum_{j=1}^{n} \omega_j p_{[j]},
\end{aligned} \tag{8}
$$

where

$$
\omega_j = \begin{cases} \alpha j + \beta(n-i)\sigma + \gamma(n+1) & 1 \le j \le k \\ \beta(n-i)\sigma + \gamma + n\delta & k < j \le l \\ \beta(n-j) + \beta(n-i)\sigma + \gamma & l < j \le i \\ \beta(n-j) + \gamma & i < j \le n \end{cases} \tag{9}
$$

It is evident that if $i = n$ no maintenance activity needs to schedule. Given the processing time $a_{[j]}$, the actual processing time $p_{[j]}$ of the scheduled $j$th job can be given as follows.

$$
p_{[j]} = a_{[j]} + b \sum_{t=1}^{m-1} (1+b)^{t-1} a_{[j-t]}, \tag{10}
$$

where $1 \le j \le n$, and

$$
m = \begin{cases} j & \text{if } j \le i \\ j-i & \text{if } j > i \end{cases} \tag{11}
$$

Combining (4), (6) and (8) and using (10), we obtain

$$
Z = M + \sum_{j=1}^{n} \omega_j p_{[j]} = M + \sum_{j=1}^{n} W_j a_{[j]}, \tag{12}
$$

where

$$
M = \begin{cases} n\mu\gamma + \alpha i \mu & i < k \\ n\delta\mu & k \le i < l \\ (n-i)\beta\mu & l \le i \le n \end{cases} \tag{13}
$$

the positional weight

$$
W_j = \omega_j + b \sum_{t=j+1}^{m'} \omega_t (1+b)^{t-j-1}, \tag{14}
$$

and

$$
m' = \begin{cases} i & \text{if } 1 \le j \le i \\ n & \text{if } i < j \le n \end{cases} \tag{15}
$$

Based on the above analysis and the rearrangement inequality (Lemma 5), we give the following algorithm to solve the problem

$$
1 \mid SLK, p_j = a_j + bt, ma \mid \sum\nolimits_{j=1}^{n} \left( \alpha E_j + \beta T_j + \gamma d_j^{(1)} + \delta D \right)
$$

**Algorithm 1**

```
1  SET k = ⌊ n(δ−γ)/α ⌋, l = ⌊ n(β−δ)/β ⌋.
2  FOR each position i = 1,2,..., n available
   to allocate a maintenance activity
3      FOR each position j = 1,2,..., n in a schedule
4          DETERMINE the positional weight W_j
5      END FOR
6      SORT the jobs in non-decreasing order of their
       normal processing times
7      ALLOCATE jobs non-decreasing ordered by their
       normal processing times to positions
       non-increasing ordered by their weights
8      DETERMINE an optimal schedule locally and its
       total cost
9  END FOR
10 DETERMINE the global optimal schedule with the
   minimum total cost
```

In Step 7, by Lemma 5, we arrange the job with the largest normal processing time to the position with the smallest value of $W_j$, the job with the second largest normal processing time to the position with the second smallest value of $W_j$, and so forth.

To this end, we obtain the following result.

**Theorem 1.** Algorithm 1 *solves the problem* $1|SLK, p_j = a_j + bt, ma|\sum_{j=1}^{n}\left(\alpha E_j + \beta T_j + \gamma d_j^{(1)} + \delta D\right)$ *in* $O\left(n^2 \log n\right)$ *time.*

**Proof:** The correction of Algorithm 1 is guaranteed by Lemmas 3-5. The time complexity of the inner loop from Step 3 to Step 8 is $O\left(n \log n\right)$. In the outer loop from Step 2 to Step 9, position index *i* takes on integer values between 1 to *n*. Hence, the time complexity for solving the $1|SLK, p_j = a_j + bt, ma|\sum_{j=1}^{n}\left(\alpha E_j + \beta T_j + \gamma d_j^{(1)} + \delta D\right)$ problem is $O\left(n^2 \log n\right)$.

## 5. Numerical Example

In this section, Algorithm 1 is illustrated by the following example.

**Example 1.** The normal processing times of $n = 9$ jobs are $a_1 = 62$, $a_2 = 81$, $a_3 = 25$, $a_4 = 82$, $a_5 = 26$, $a_6 = 19$, $a_7 = 55$, $a_8 = 9$ and $a_9 = 91$. The overall cost consists of four specific costs for unit earliness $\alpha = 4$, unit tardiness $\beta = 15$, unit due-window size $\delta = 6$ and unit due-window starting time $\gamma = 5$. Set the common deteriorating factor $b = 0.05$, the deteriorating maintenance factor $\sigma = 0.1$, and the basic maintenance time $\mu = 10$.

*Solution*: As shown in Step 1 in Algorithm 1, we determine

$$k = \left\lfloor \frac{n(\delta - \gamma)}{\alpha} \right\rfloor = 2 \text{ and } l = \left\lfloor \frac{n(\beta - \delta)}{\beta} \right\rfloor = 5.$$

As shown in Table 1, all the local optimal job sequences and their total costs are presented, among which the optimal total cost is underlined. The global optimal solution for this example is illustrated as: 1) the job sequence is (7, 8, 6, 3, 5, 1, 2, 4, 9) and the corresponding job starting time and actual processing time are (0.00, 70.50, 79.50, 98.95, 125.37, 154.12, 220.30, 308.79, 402.70) and

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

| $i$ | Job sequence | $Z$ |
|---|---|---|
| 1 | (7, 8, 6, 3, 5, 1, 2, 4, 9) | 17,476.37 |
| 2 | (7, 5, 8, 6, 3, 1, 2, 4, 9) | 17,525.07 |
| 3 | (3, 6, 7, 8, 5, 1, 2, 4, 9) | 17,634.66 |
| 4 | (6, 8, 3, 7, 5, 1, 2, 4, 9) | 17,749.44 |
| 5 | (6, 8, 3, 5, 7, 1, 2, 4, 9) | 18,157.92 |
| 6 | (6, 8, 3, 5, 7, 1, 2, 4, 9) | 18,271.87 |
| 7 | (6, 8, 3, 5, 7, 1, 2, 4, 9) | 18,347.63 |
| 8 | (6, 8, 3, 5, 7, 1, 2, 4, 9) | 18,170.85 |
| 9 | (6, 8, 3, 5, 7, 1, 2, 4, 9) | 17,519.13 |

(55.00, 9.00, 19.45, 26.42, 28.74, 66.18, 88.49, 93.91, 107.61), respectively; 2) the slack window parameters are $q^{(1)} = 79.50$ and $q^{(2)} = 154.12$; 3) the maintenance activity is located right after the first job (*i.e.* Job 7), starting at time $t = 55.00$ and ending at time $t = 70.50$; 4) the total cost is $Z = 17476.37$.

## 6. Conclusion

We solved a single machine slack due-window assignment and scheduling problem of a deteriorating maintenance activity and linear time-dependent deteriorating jobs, and gave a polynomial-time algorithm. The running time of this algorithm does not exceed $O\left(n^2 \log n\right)$. The problem with the setting of parallel identical machines, or the problems with min-max type objective functions may be considered in the future.


## Fund

This work was supported by NSFGD (2018A030313267).


## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References


[1] Cheng, T.C.E. (1988) Optimal Common Due-Date with Limited Completion Time Deviation. *Computers & Operations Research*, **15**, 91-96. https://doi.org/10.1016/0305-0548(88)90001-9

[2] Liman, S., Panwalkar, S. and Thongmee, S. (1998) Common Due Window Size and Location Determination in a Single Machine Scheduling Problem. *Journal of the Operational Research Society*, **49**, 1007-1010. https://doi.org/10.1057/palgrave.jors.2600601

[3] Mosheiov, G. and Sarig, A. (2008) A Multi-Criteria Scheduling with Due-Window Assignment Problem. *Mathematical and Computer Modelling*, **48**, 898-907.

https://doi.org/10.1016/j.mcm.2007.08.018

[4] Kang, L.Y., Cheng, T.C.E., Ng, C.T. and Zhao, M. (2005) Scheduling to Minimize Makespan with Time-Dependent Processing Times. *Lecture Notes in Computer Science*, **3827**, 925-933. https://doi.org/10.1007/11602613_92

[5] Wang, J.-B. and Wang, C. (2011) Single-Machine Due-Window Assignment Problem with Learning Effect and Deteriorating Jobs. *Applied Mathematical Modelling*, **35**, 4017-4022. https://doi.org/10.1016/j.apm.2011.02.023

[6] Yin, N., Kang, L.Y., Ji, P. and Wang, J.B. (2014) Single Machine Scheduling with Sum-of-Logarithm-Processing-Times Based Deterioration. *Information Sciences*, **274**, 303-309. https://doi.org/10.1016/j.ins.2014.03.004

[7] Lee, C.-Y. and Leon, V. (2001) Machine Scheduling with a Rate-Modifying Activity. *European Journal of Operational Research*, **128**, 119-128. https://doi.org/10.1016/S0377-2217(99)00066-1

[8] Zhao, C.-L. and Tang, H.-Y. (2010) Single Machine Scheduling with General Job-Dependent Aging Effect and Maintenance Activities to Minimize Makespan. *Applied Mathematical Modelling*, **34**, 837-841. https://doi.org/10.1016/j.apm.2009.07.002

[9] Cheng, T.C.E., Yang, S.-J. and Yang, D.-L. (2012) Common Due-Window Assignment and Scheduling of Linear Time-Dependent Deteriorating Jobs and a Deteriorating Maintenance Activity. *International Journal of Production Economics*, **135**, 154-161. https://doi.org/10.1016/j.ijpe.2010.10.005

[10] Zhao, C.-L. and Tang, H.-Y. (2012) A Note to Due-Window Assignment and Single Machine Scheduling with Deteriorating Jobs and a Rate-Modifying Activity. *Computers & Operations Research*, **39**, 1300-1303. https://doi.org/10.1016/j.cor.2010.04.006

[11] Ji, M., Ge, J., Chen, K. and Cheng, T.E. (2013) Single-Machine Due-Window Assignment and Scheduling with Resource Allocation, Aging Effect, and a Deteriorating Rate-Modifying Activity. *Computers & Industrial Engineering*, **66**, 952-961. https://doi.org/10.1016/j.cie.2013.08.020

[12] Cheng, B. and Cheng, L. (2014) Note on Single-Machine Due-Window Assignment and Scheduling with Resource Allocation, Aging Effect, and a Deteriorating Rate-Modifying Activity? *Computers & Industrial Engineering*, **78**, 320-322. https://doi.org/10.1016/j.cie.2014.07.013

[13] Mosheiov, G. and Oron, D. (2010) Job-Dependent Due-Window Assignment Based on a Common Flow Allowance. *Foundations of Computing and Decision Sciences*, **35**, 185-195.

[14] Hardy, G.H., Littlewood, J.E. and Pólya, G. (1934) Inequalities. 1em plus 0.5em minus 0.4em. Cambridge University Press.