Single-machine scheduling with piece-rate maintenance and interval constrained position-dependent processing times

Pengfei Xue a, Yulin Zhang a,⇑, Xianyu Yu a,b

a School of Economics and Management, Southeast University, Nanjing 210096, China
b School of Science, East China Institute of Technology, Fuzhou, Jiangxi 344000, China

Keywords:
Single-machine
Scheduling
Position-dependent
Maintenance
Total cost

Abstract
This paper investigates single-machine scheduling problems with piece-rate machine maintenance and interval constrained actual processing time. The actual processing time of a job is a general function of the normal job processing time and the position in job sequence, and it is required to restrict in given interval otherwise earliness or tardiness penalty should be paid. The maintenance duration studied in the paper is a time-dependent linear function. The objective is to find jointly the optimal frequency to perform maintenance and the optimal job sequence to minimize the total cost, which is a linear function of the makespan, total earliness and total tardiness. There is shown that the problem can be optimally solved in $O(n^4)$ time. There is also shown that two special cases of the problem can be optimally solved by lower order algorithms.

1. Introduction

Recent developments in the field of job scheduling have triggered into a growing interest towards learning or aging effect. In case of the learning effect, the actual processing time of a job will be shorter when it is scheduled later in a sequence. While in case of the aging effect, the actual processing time of a job will be longer when it is scheduled later in a sequence. Some papers discussed time-dependent scheduling problems. The relevant papers include Bachman et al. [1], Cheng et al. [2] and Ji and Cheng [3]. Mosheiov and Oron [4] and Hsu et al. [5] discussed position-dependent scheduling problems. For scheduling problems with position-dependent effect, the position-dependent function as specific linear or exponential function was considered by Cheng and Wang [6], Bachman and Janiaik [7], Wang and Xia [8], Biskup [9], Kuo and Yang [10], Yang, Yang and Cheng [11], Yang and Yang [12] and Zhao and Tang [13]. Some relevant references for the combination of the exponential function and linear function are: Lee [14], Cheng, Wu and Lee [15], Wang [16], Wang and Cheng [17], Cheng and Lee [18], Yin et al. [19] and Lai and Lee [20]. However, all these papers consider learning or aging model with specific processing time function. Only a few papers study general position-dependent processing times. Mosheiov and Sidney [21] and Wang and Chong [22] considered a scheduling problem with a general position-dependent learning function. Mosheiov [23] studied a scheduling problem with position-dependent job processing times which is not restricted to a monotone function.

In addition, it is well-known that the production efficiency can be improved by performing maintenance in manufacturing processes. Some scheduling researchers considered that the machine is maintained exactly once during the planning period, and the duration of maintenance is not fixed. Some recent relevant references are: Kubzin and Strusevich [24], Mosheiov and Sidney [25], Cheng et al. [26], Mor and Mosheiov [27] and Cheng et al. [28]. Some scheduling researchers assumed that the duration of each maintenance is given. Liao and Chen [29] and Ji et al. [30] considered single-machine scheduling...
problems with periodic maintenance activities, where each maintenance activity is required after a periodic time interval. Ji and Cheng [31] and Yang et al. [32] assumed that the machine may have multiple rate-modifying activities over the scheduling horizon.

Another popular topic in recent years is scheduling problem with job completion time due window. To cope with global competition and to improve customer demand, jobs should be finished as close as possible to their due-dates. If a job is finished earlier than its due-date, it would result in storage costs; on the other hand, if a job is finished later than its due-date, it would violate contractual obligation with the customer. Cheng [33] initiated research on scheduling with due windows. In modern manufacturing management, scheduling problems with due window assignment are important issues. Many papers have been conducted on these issues in different scheduling environments. Yin et al. [34] studied a single-machine scheduling problem with batch delivery cost and an assignable common due window simultaneously. They provided polynomial-time solutions for considered problems. Yeung et al. [35] studied a non-preemptive two-stage flowshop scheduling problem under the environment of a common due window. The objective is to minimize the earliness and tardiness. They showed that the problem is NP-complete in the strong sense, and developed a branch and bound algorithm and a heuristic to solve the problem. For more recent works on scheduling with a common due window, we refer the reader to the comprehensive surveys by Cheng and Gupta [36], Gordon et al. [37], and the recent papers by Yeung et al. [38], Cheng [39], Baker and Scudder [40], Liman et al. [41], Yeung et al. [42], Yeung [43], Mosheiov and Sarig [44], Yeung45, Yang et al. [46], Yin et al. [47] and Yin et al. [48]. In this paper, we assume that every job has its own interval constrained actual processing time, which is required to restrict in given interval otherwise earliness or tardiness penalty should be paid. For example, in porcelain manufacturing processes, the actual processing time cannot exceed a given interval otherwise the porcelain may have quality flaws.

To the best of our knowledge, however, scheduling with simultaneous consideration of interval constrained actual processing time and machine maintenance has not been explored. Motivated by these points, in this paper we consider the single machine scheduling with interval constrained actual processing time and piece-rate maintenance, where the piece-rate maintenance is proposed by Yu et al. [49]. We explore to find the optimal polynomial algorithm to minimize the total cost, which is assumed to conclude production fee and total earliness and tardiness cost. Moreover, we investigate two special cases where the actual processing time of each job is assumed to be a product function, and explore to find more effective solving algorithm to minimize the total cost for the cases.

This paper is organized as follows. We introduce the notation and terminology of this paper in the next section. In Section 3, we propose the main results of this paper. In Section 4, we conclude this paper, and suggest some topics for future.

2. Notations and problem formulation

A set $J = \{J_1, J_2, \ldots, J_n\}$ of $n$ independent jobs are partitioned into $m + 1$ groups and are all available for processing at time zero. The machine can handle one job at a time. In the manufacturing process, the jobs are non-preemptive. Each job $j \in J$ is associated with the normal processing time $p_j$. We extend the general processing time function of [23] and construct a more general model of the actual job processing time

$$p_{j} = f(p_{j}, r), \quad (1 \leq j \leq n, \quad 1 \leq r \leq n).$$

Observing from Eq. (1), different jobs may have different actual processing time functions which are not restricted to be specific and monotone.

During the maintenance, the machine is stopped. After the maintenance activity, the machine will revert to its initial condition. We assume that the machine need to be maintained one time when every $L$ jobs are completed and maintenance is just finished at time zero. The jobs will be processed from a group continuously. Thus, we denote the schedule as $\sigma = [G_1, M_1, \ldots, G_m, M_m, G_{m+1}]$, where $G_i$ denotes the $i$th group and $M_i$ denotes the $i$th maintenance. Let $n_i$ denote the number of jobs in the $i$th group, and $m$ be the maintenance frequency, which is equal to the integer part of $\frac{L}{n}$, i.e., $m = \lfloor \frac{L}{n} \rfloor$. We can obtain that $\sum_{i=1}^{m} n_i = n$. Let $C_{j;r}$ be the completion time of the job scheduled in $r$th position of the $l$th group.

Let $G_i$ denote the time length of $i$th group, and $t_i$ be the duration of the maintenance, i.e., $t_i = aG_i + b$, where $a$ and $b$ are two constants, $a \geq 0$. As in Zhao and Tang [13], we denote the group $G_i$ as $\tilde{G}_i = [J_{i,1}; J_{i,2}; \ldots; J_{i,n_i-1}; J_{i,n_i}]$, where $J_{i,j}$ is the $j$th job in the group $G_i$.

Let $p_{j;\tau}$ denote the normal processing time of job $J_{i,j}$ and $p_{j;\tau}$ be actual processing time of job $J_{i,j}$, where $p_{j;\tau} = f(p_{j;\tau}, r)$. Let $a_i$ and $b_i$ denote the lower limit and upper limit of the interval constraint of every job, respectively. Then, the earliness of job $j$ is denoted as $E_{j;\tau}$, i.e., $E_{j;\tau} = \max(0, a_j - p_{j})$. The tardiness of job $j$ is denoted as $T_{j;\tau}$, i.e., $T_{j;\tau} = \max(0, p_{j} - b_j)$. Let $C_{\text{max}}$ denote the makespan, i.e., $C_{\text{max}} = \max(C_{j;\tau} | j = 1, 2, \ldots, n)$, where $C_{j;\tau}$ denotes the completion time of the job $j$. The total earliness of all jobs is denoted by $\sum_{j=1}^{n} E_{j;\tau}$, and the total tardiness of all jobs is denoted by $\sum_{j=1}^{n} T_{j;\tau}$.

In the manufacturing process, the production fee is determined by the length of the working time. Moreover, the earliness cost and tardiness cost are assumed to be linear relationship with the total earliness and tardiness of all jobs, respectively. Thus, We define the total cost as follows:

$$TC = \alpha C_{\text{max}} + \beta \sum_{j=1}^{n} E_{j;\tau} + \gamma \sum_{j=1}^{n} T_{j;\tau}.$$
where α, β and γ are the unit production fee, the unit earliness cost and the unit tardiness cost, respectively. α, β and γ should be positive numbers, i.e., α, β, γ > 0.

3. The main results

The completion time of all jobs can be obtained as follows:

\[ C_{[m+1,n_{m-1}]} = (a+1) \sum_{j=1}^{m-1} \sum_{j=1}^{n_j} f_{j_{p_j}}(p_{j_{p_j}}, r) + \sum_{j=1}^{m-1} f_{m+1,r}(p_{m+1,r}, r) + m b. \]  

From the Eq. (2), it can be obtained that

\[ C_{\max} = C_{[m+1,n_{m-1}]} = (a+1) \sum_{j=1}^{m-1} \sum_{j=1}^{n_j} f_{j_{p_j}}(p_{j_{p_j}}, r) + \sum_{j=1}^{m-1} f_{m+1,r}(p_{m+1,r}, r) + m b. \]

The total earliness is given by

\[ \sum_{j=1}^{m} \sum_{j=1}^{n_j} E_{j_{p_j}} = \sum_{j=1}^{m} \sum_{j=1}^{n_j} \max \{0, a_{j_{p_j}} - f_{j_{p_j}}(p_{j_{p_j}}, r)\} + \sum_{j=1}^{m} \sum_{j=1}^{n_j} \max \{0, a_{j_{p_j}} - f_{j_{p_j}}(p_{j_{p_j}}, r)\}. \]

Similarly, the total tardiness is given by

\[ \sum_{j=1}^{m} \sum_{j=1}^{n_j} T_{j_{p_j}} = \sum_{j=1}^{m} \sum_{j=1}^{n_j} \max \{0, f_{j_{p_j}}(p_{j_{p_j}}, r) - b_{j_{p_j}}\} + \sum_{j=1}^{m} \sum_{j=1}^{n_j} \max \{0, f_{j_{p_j}}(p_{j_{p_j}}, r) - b_{j_{p_j}}\}. \]

Then, the total cost can be obtained as follows:

\[ TC = \alpha C_{\max} + \beta \sum_{j=1}^{m} \sum_{j=1}^{n_j} E_{j_{p_j}} + \gamma \sum_{j=1}^{m} \sum_{j=1}^{n_j} T_{j_{p_j}} = \alpha \left( \sum_{j=1}^{m} \sum_{j=1}^{n_j} (a+1) f_{j_{p_j}}(p_{j_{p_j}}, r) + m b + \sum_{j=1}^{m} \sum_{j=1}^{n_j} f_{m+1,r}(p_{m+1,r}, r) \right) \]
\[ + \beta \left( \sum_{j=1}^{m} \sum_{j=1}^{n_j} \max \{0, a_{j_{p_j}} - f_{j_{p_j}}(p_{j_{p_j}}, r)\} + \sum_{j=1}^{m} \sum_{j=1}^{n_j} \max \{0, a_{j_{p_j}} - f_{j_{p_j}}(p_{j_{p_j}}, r)\}\right) \]
\[ + \gamma \left( \sum_{j=1}^{m} \sum_{j=1}^{n_j} \max \{0, f_{j_{p_j}}(p_{j_{p_j}}, r) - b_{j_{p_j}}\} + \sum_{j=1}^{m} \sum_{j=1}^{n_j} \max \{0, f_{j_{p_j}}(p_{j_{p_j}}, r) - b_{j_{p_j}}\}\right) \]
\[ - \sum_{j=1}^{m} \sum_{j=1}^{n_j} \left( \alpha (a+1) f_{j_{p_j}}(p_{j_{p_j}}, r) + \beta \max \{0, a_{j_{p_j}} - f_{j_{p_j}}(p_{j_{p_j}}, r)\} + \gamma \max \{0, f_{j_{p_j}}(p_{j_{p_j}}, r) - b_{j_{p_j}}\}\right) \]
\[ + \sum_{j=1}^{m} \sum_{j=1}^{n_j} \left( \alpha f_{m+1,r}(p_{m+1,r}, r) + \beta \max \{0, a_{m+1,r} - f_{m+1,r}(p_{m+1,r}, r)\} + \gamma \max \{0, f_{m+1,r}(p_{m+1,r}, r) - b_{m+1,r}\}\right) + \alpha m b. \]

For simplicity, let \( \beta \max \{0, a_{j_{p_j}} - f_{j_{p_j}}(p_{j_{p_j}}, r)\} + \gamma \max \{0, f_{j_{p_j}}(p_{j_{p_j}}, r) - b_{j_{p_j}}\} \) be denoted by \( Z \). Then, we introduce a new function \( f_{j_{p_j}}^{(1)}(p_{j_{p_j}}, r) \) as follows:

\[ f_{j_{p_j}}^{(1)}(p_{j_{p_j}}, r) = \begin{cases} 
\alpha (a+1) f_{j_{p_j}}(p_{j_{p_j}}, r) + Z, & l = 1, 2, \ldots, m, \\
\gamma f_{j_{p_j}}(p_{j_{p_j}}, r) + Z, & l = m + 1.
\end{cases} \]

Combining (6) and (7), we can obtain

\[ TC_{[m+1,n_{m-1}]} = \sum_{l=1}^{m+1} \sum_{r=1}^{n_i} f_{j_{p_j}}^{(1)}(p_{j_{p_j}}, r) + \alpha m b. \]

The problem we studied in this section can be denoted by \( 1 | p_{j_{p_j}} | f_{j_{p_j}}(p_{j_{p_j}}, r), PRM = k, t_i = a G_t^j + b | TC \). For a given the number of jobs \( k \), we can get the number of groups by equation \( m = \left\lfloor \frac{k}{2} \right\rfloor \). Then, we can obtain that \( m b \) is a constant. We explore to find a polynomial to solve the problem. The problem \( 1 | p_{j_{p_j}} | f_{j_{p_j}}(p_{j_{p_j}}, r), PRM = k, t_i = a G_t^j + b | TC \) can be transformed as the following standard assignment problem:

\[
\begin{array}{c}
\min \sum_{j=1}^{m} \sum_{l=1}^{n_i} \sum_{r=1}^{n_j} w_{j_{p_j}} x_{j_{p_j}} + \alpha m b, \\
st. \sum_{j=1}^{n_i} x_{j_{p_j}} = 1, & l = 1, 2, \ldots, m + 1, \quad r = 1, 2, \ldots, n_i, \\
\sum_{l=1}^{m+1} \sum_{r=1}^{n_i} x_{j_{p_j}} = 1, & j = 1, 2, \ldots, n, \\
x_{j_{p_j}} = 0 \text{ or } 1 & j = 1, 2, \ldots, n, \quad l = 1, 2, \ldots, m + 1, \quad r = 1, 2, \ldots, n_i,
\end{array}
\]

where \( w_{j_{p_j}} = f_{j_{p_j}}^{(1)}(p_{j_{p_j}}, r) \). We define \( x_{j_{p_j}} = 0 \) or 1 such that \( x_{j_{p_j}} = 1 \) if job \( J_j \) is scheduled in the \( r \)th position in the group \( G_i \) and \( x_{j_{p_j}} = 0 \) otherwise. In case of \( k = n \), all jobs are completed in one group, and the machine does not require maintenance. Thus, the objective of the assignment problem is not \( \sum_{j=1}^{n} \sum_{l=1}^{m+1} \sum_{r=1}^{n_i} w_{j_{p_j}} x_{j_{p_j}} + \alpha m b \), but \( \sum_{j=1}^{n} \sum_{l=1}^{m+1} \sum_{r=1}^{n_i} w_{j_{p_j}} x_{j_{p_j}} \).

In order to minimize the total cost, we propose a polynomial time algorithm to determine jointly the optimal \( k \), and the optimal job sequence.

**Algorithm 1.**

Step 1. For each \( k (k=1,2,\ldots,n-1,n) \), solve the assignment problem (8) and calculate the objective value \( TC(k) \).

Step 2. Let \( (TC(k'))=\min (TC(k), (k=1,2,\ldots,n)) \).
Theorem 1. The 11|pj = fj(p, r). PRM = k, tj = acj + b/TC problem can be optimally solved by Algorithm 1 in O(n^4) time.

Proof. For a fixed number of jobs k, the problem 1|pj = fj(p, r). PRM = k, tj = acj + b/TC can be optimally solved via the assignment problem (8), and the computational complexity of obtaining the optimal job sequence is O(n^4). Since k has n possible values, the computational complexity of solving the problem 1|pj = fj(p, r). PRM = k, tj = acj + b/TC is O(n^4).

In what follows, we investigate two special cases of the 1|pj = fj(p, r). PRM = k, tj = acj + b/TC problem, and explore to find a more efficient algorithm for each case. We assume that \( p_j^l = p_j r^{a0} \) (i.e., \( p_j^l = p_j r^{a0} \)) and \( t_j = b(b > 0) \) and \( x \geq b \).

The first case concerns the position-dependent learning effect, \( a_0 < 0 \). In this case, we assume that every job has its own constrained interval and has no tardiness. Let \( p_j b_0 (0 < b_0 < 1) \) and \( p_j \) denote the lower limit and upper limit of the constrained interval of the job \( j \), respectively.

The second case concerns the position-dependent aging effect, \( a_0 > 0 \). In this case, we assume that every job has its own constrained interval and has no earliness. Let \( p_j \) and \( p_j b_0 (b_0 > 1) \) denote the lower limit and upper limit of the constrained interval of the job \( j \), respectively.

We first consider the case of the position-dependent learning effect. It is obviously that machine does not need maintenance in the optimal scheduling. Then, we can denote the scheduling problem with learning effect as 11|pj = pj r^{a0}|TC.

Theorem 2. If \( a_0 < 0 \), the problem 1|pj = pj r^{a0}|TC can be optimally solved by the shortest processing time first (SPT) rule in O(n log n) time.

Proof. Assume that an optimal schedule \( \sigma_1 \) is not the SPT sequence. Then, there must exist a pair of adjacent jobs \( i \) and \( j \) such that \( p_i > p_j \), with job \( j \) following job \( i \). We assume that job \( i \) is scheduled in the \( r \)th position.

Now let a new job schedule \( \sigma_2 \) be formed from \( \sigma_1 \) by interchanging jobs \( i \) and \( j \) and keeping the other jobs in the same position as in \( \sigma_1 \). The exchange of the jobs \( i \) and \( j \) is illustrated by Fig. 1, where \( A \) denotes the set of jobs preceding jobs \( i \) and \( j \) in both schedules, and \( B \) is the set of jobs following jobs \( i \) and \( j \) in both schedules. Let \( TC(\sigma_1) \) and \( TC(\sigma_2) \) denote the total cost of \( \sigma_1 \) and \( \sigma_2 \), respectively. Since the positions of the jobs in \( A \) and \( B \) remain unchanged, the cost of processing and the earliness of the jobs in \( A \) and \( B \) remains unchanged.

Since not all jobs in the sequence have earliness in case of learning effect, we aim to find the tipping position \( r^* \), whose illustration is proposed in Fig. 2. Let \( p_j b_0 = p_j r^{a0} \), we can obtain \( r^* = \lceil \sqrt{b_0} \rceil \). In case of \( r \leq r^* \), since \( p_j b_0 \leq p_j r^{a0} \), it can be seen that the job before the position \( (r^* + 1) \) has no earliness. While in case of \( r > r^* \), since \( p_j b_0 > p_j r^{a0} \), it can be seen that the job behind the position \( r^* \) has earliness. In order to prove Theorem 2, we consider the following three cases:

(1) \( r < r^* \);
(2) \( r = r^* \);
(3) \( r > r^* \).

Case (1). \( r < r^* \)

In sequence \( \sigma_1 \) and \( \sigma_2 \), the jobs \( i \) and \( j \) have no earliness. Then, it can be obtained

\[
TC(\sigma_1) - TC(\sigma_2) = \alpha(p_j r^{a0} + p_j (r + 1)^{a0}) - \alpha(p_j r^{a0} + p_j (r + 1)^{a0}) = \alpha(p_j - p_j)(r^{a0} - (r + 1)^{a0}) \geq 0.
\]

Then, the schedule \( \sigma_1 \) is dominated by the schedule \( \sigma_2 \) in this case.

Case (2). \( r = r^* \)

In sequence \( \sigma_1 \), the job \( i \) has no earliness and the job \( j \) has earliness, but in the sequence \( \sigma_2 \) the case is converse. Then, it can be obtained

\[
TC(\sigma_1) - TC(\sigma_2) = (\alpha(p_j r^{a0} + p_j (r + 1)^{a0}) + \beta(p_j b_0 - p_j (r + 1)^{a0}) - (\alpha(p_j r^{a0} + p_j (r + 1)^{a0}) + \beta(p_j b_0 - p_j (r + 1)^{a0}))
= (p_j - p_j)(\alpha(r^{a0} - (r + 1)^{a0}) - \beta(b_0 - (r + 1)^{a0})) \geq 0.
\]

Then, the schedule \( \sigma_1 \) is dominated by the schedule \( \sigma_2 \) in this case.
Case (3). \( r > r^* \)

In sequence \( \sigma_1 \) and \( \sigma_2 \), the jobs \( i \) and \( j \) have earliness. Then, it can be obtained

\[
TC(\sigma_1) - TC(\sigma_2) = (\alpha(p_tr^{\alpha_0} + p_j(r + 1)^{\alpha_0}) + \beta(p_0b_0 - p_j(r + 1)^{\alpha_0}) - (\alpha(p_tr^{\alpha_0} + p_j(r + 1)^{\alpha_0}) + \beta(p_0b_0 - p_j(r + 1)^{\alpha_0})) - (\alpha(p_tr^{\alpha_0} + p_j(r + 1)^{\alpha_0}) + \beta(p_0b_0 - p_j(r + 1)^{\alpha_0}) = (r^{\alpha_0} - (r + 1)^{\alpha_0})p_j(\alpha - \beta) \geq 0.
\]

Then, the schedule \( \sigma_1 \) is dominated by the schedule \( \sigma_2 \) in this case.

Summing the above argument, in any case, we obtain that the interchange of jobs \( i \) and \( j \) will result in a decrease in total cost. By the repeated application of this argument, any sequence that is not a \( SPT \) sequence can be improved with respect to the total cost. Then, the problem \( 1|p_j = p_tr^{\alpha_0}|TC \) can be optimally solved by the \( SPT \) rule in \( O(n \log n) \) time. \( \square \)

In what follows, we discuss the case of the position-dependent aging effect. It is necessary to perform maintenance for the machine in the optimal schedule of the \( 1|p_j = p_tr^{\alpha_0}, \text{PRM} = k, t_i = b|TC \) problem. We present a modified Longest Processing Time first (LPT) rule for the \( 1|p_j = p_tr^{\alpha_0}, \text{PRM} = k, t_i = b|TC \) problem.

The modified LPT rule. At first, we sequence the jobs in non-increasing order of their normal processing times. Then, schedule the job in the first position of each group one by one. If the first position of each group is filled, then schedule the remaining job in the second position of each group one by one. If all the second positions are filled, fill the third position, and so on, until all jobs are scheduled.

**Theorem 3.** If \( a_0 > 0 \), the problem \( 1|p_j = p_tr^{\alpha_0}, \text{PRM} = k, t_i = b|TC \) can be optimally solved by the modified LPT rule in \( O(n^2 \log n) \) time.

**Proof.** Assume that an optimal schedule \( \sigma_3 \) is not sequenced by the modified LPT rule. We consider jobs \( J_{i|x} \) and \( J_{j|x+1} \). Job \( J_{j|x} \) is sequenced in \( r \)th position of the group \( G_i \). Job \( J_{j|x+1} \) is sequenced in \( (r + 1) \)th position of the group \( G_j \). \( p_{i|x} \) and \( p_{j|x+1} \) denote the normal processing times of jobs \( J_{i|x} \) and \( J_{j|x+1} \), respectively. Moreover, assume that \( p_{i|x} \leq p_{j|x+1} \).

Now let a new schedule \( \sigma_4 \) be formed from \( \sigma_3 \) by interchanging jobs \( J_{i|x} \) and \( J_{j|x+1} \) and keeping the other jobs in the same position as in \( \sigma_3 \). The exchange of the jobs \( J_{i|x} \) and \( J_{j|x+1} \) is illustrated by Fig. 3, where \( \pi_1, \pi_2 \) and \( \pi_3 \) are partial schedule of the schedule \( \sigma_3 \). Let \( TC(\sigma_3) \) and \( TC(\sigma_4) \) denote the total cost of \( \sigma_3 \) and \( \sigma_4 \), respectively. For the reason that the positions of the jobs in \( \pi_1, \pi_2 \) and \( \pi_3 \) unchanged, the cost of processing and the tardiness of every job in \( \pi_1, \pi_2 \) and \( \pi_3 \) remains unchanged.

Since not all jobs in the sequence have tardiness in case of aging effect, so we aim to find the tipping position \( r^* \), whose illustration is proposed in Fig. 4. Let \( p_0b_0 = p_tr^{\alpha_0} \), we can obtain \( r^* = \lfloor \sqrt[\beta]{b_0} \rfloor \). In case of \( r \leq r^* \), since \( p_0b_0 \geq p_tr^{\alpha_0} \), the job before the position \( r^* + 1 \) has no tardiness. While in case of \( r > r^* \), since \( p_0b_0 < p_tr^{\alpha_0} \), the job behind the position \( r^* \) has tardiness. In order to prove **Theorem 3**, we consider the following three cases:

1. \( r < r^* \);
2. \( r = r^* \);
3. \( r > r^* \).
case (1). $r < r^*$

In sequence $\sigma_3$ and $\sigma_4$, the jobs $J_{ir}$ and $J_{ir+1}$ have no tardiness. Then, it can be obtained

$$TC(\sigma_3) - TC(\sigma_4) = \alpha(p_{ir}^r + p_{ir+1}^r) - \alpha(p_{ir+1}^r + p_{ir}^r) = \alpha(r^0 - (r + 1)^0) \geq 0.$$  

Then, the schedule $\sigma_3$ is dominated by the schedule $\sigma_4$.

case (2). $r = r^*$

In sequence $\sigma_3$, the job $J_{ir}$ has no tardiness and the job $J_{ir+1}$ has tardiness, but in the sequence $\sigma_4$ the case is converse. Then, it can be obtained

$$TC(\sigma_3) - TC(\sigma_4) = \alpha(p_{ir}^r - p_{ir+1}^r + p_{ir+1}^r - p_{ir}^r) = \alpha(r^0 - (r + 1)^0) \geq 0.$$  

Then, the schedule $\sigma_3$ dominates the schedule $\sigma_3$ in this case.

case (3). $r > r^*$

In sequence $\sigma_3$ and $\sigma_4$, the jobs $J_{ir}$ and $J_{ir+1}$ have tardiness. Then, it can be obtained

$$TC(\sigma_3) - TC(\sigma_4) = \alpha(p_{ir}^r + p_{ir+1}^r - p_{ir+1}^r + p_{ir}^r) = \alpha(r^0 - (r + 1)^0) \geq 0.$$  

Then, the schedule $\sigma_3$ is dominated by the schedule $\sigma_4$ in this case.

Based on the above argument, we obtain that the interchange of jobs $J_{ir}$ and $J_{ir+1}$ will result in a decrease in total cost. By the repeated application of this argument, any sequence that is not a modified $LPT$ sequence can be improved with respect to the total cost. For a fixed $k$, the $1|p_j^r = p_j^r| \text{PRM} - k, t_i = b_i|TC$ problem can be optimally solved by the modified $LPT$ rule, and the computational complexity is $O(n \log n)$. Since $k(k = 1, 2, \ldots, n)$ has $n$ possible values, the total computational complexity of solving the problem $1|p_j^r = p_j^r| \text{PRM} - k, t_i = b_i|TC$ is $O(n^2 \log n)$. □

4. Conclusions

The paper considered the scheduling problem with piece-rate machine maintenance and interval constrained actual processing time. The job actual processing time is required to restrict in given interval otherwise earliness or tardiness penalty should be paid. The objective is to minimize the total cost that is a linear function of the makespan, the earliness and tardiness penalties of all jobs. For scheduling problem with general position-dependent processing times, we showed that the total cost minimization problem can be optimally solved in $O(n^4)$ time. Moreover, for the two special cases, we showed that the total cost minimization problem with learning effect can be solved by $SPT$ rule in $O(n \log n)$ time, and the total cost minimization problem with aging effect can be solved by the modified $LPT$ rule in $O(n^2 \log n)$ time. For the future research, it is suggested to investigate the scheduling problem with piece-rate machine maintenance and interval constrained actual processing time in the context of parallel machines scheduling problems or job-shop scheduling problems.

Acknowledgements

The authors wish to thank editors and anonymous referees for their helpful comments. This work is supported by National Nature Science Foundation Project of China (71171046) and the Scientific Research Foundation of Graduate School of Southeast University (YBJJ1239).

References


