Applied Mathematical Modelling 000 (2016) 1–10

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Single-machine scheduling problems with machine aging effect and an optional maintenance activity

Manzhan Gu[a,e], Xiwen Lu[b], Jinwei Gu[c,f,*], Ying Zhang[d]

[a] School of Mathematics, Shanghai University of Finance and Economics, Shanghai 200433, China
[b] Department of Mathematics, East China University of Science and Technology, Shanghai 200237, China
[c] College of Economics and Management, Shanghai University of Electric Power, Shanghai 200090, China
[d] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[e] School of Mathematics and Statistics, Shandong University, Weihai, Shandong 264209, China
[f] School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai, Shandong 264209, China

ARTICLE INFO

Article history:
Received 16 October 2014
Revised 23 December 2015
Accepted 12 January 2016
Available online xxx

Keywords:
Aging effect
Maintenance
Makespan
Total completion times

ABSTRACT

This paper considers two single-machine scheduling problems with a new type of aging effect, which is dominated by the processing speed of the machine. During the whole scheduling horizon, the machine is subject to an optional maintenance, and the duration of the maintenance depends on the length of the uptime before it. The objective is to schedule all jobs and find the location of the maintenance so as to minimize the makespan or the total completion times. The two problems are proved to be NP-complete, and two dynamic programming algorithms are proposed to solve the problems. We analyze the computation complexity of the algorithms, and show that the problems under study are solvable in polynomial time if the processing loads of all jobs are uniformly bounded.

1. Introduction

In the traditional single-machine scheduling problems, it is assumed that the machine is available from time zero to the completion time of the last job. However, in many practical settings the machine may not be available because of the need of repair, cooling or other maintenance operations. Therefore, it is more reasonable to consider the downtimes (i.e., maintenance activities) in the scheduling problems. Scheduling under such machine environment is called as the model with availability constraints [1]. In this model, the machine is unavailable to process jobs during the period of maintenance. After the maintenance, the machine will recover to its initial state and start anew. In the past decade, the scheduling problems with maintenance activities has received more and more attention. Schmidt [2] proposed a survey on different models and problems with available constrains, and an updated survey was presented by Ma et al. [3].

In addition, In the majority of the literature, it is assumed that the processing times of jobs are known constants. However, there are many settings in which the actual processing time of each job may be affected by its position or start time in the schedule. The phenomenon is defined as the time-dependent or position-dependent aging effect in the scheduling problems. In the model with the aging effect, in a common uptime, the later a job is processed in the schedule, the longer its actual processing time is.

* Corresponding author. Tel.: +86 18918307617.
E-mail address: gujinwei1982@163.com (J. Gu).

http://dx.doi.org/10.1016/j.apm.2016.01.038

Gupta and Gupta [4] firstly studied the problem with aging effect. Since then, scheduling problems with aging effect have received the attention of many researchers, including Bachman and Janiak [5], Mosheiov [6], and Yang and Wang [7], among others. For a comprehensive review of the results in this field, the reader may refer to the recent surveys [3,5,6,8–12].

This paper considers the single-machine scheduling with aging effect and available constrains. Browne and Yechiali [13] studied the scheduling problem with time-dependent aging effect, i.e., the actual processing time of job j is $\alpha_j s_j$, where α_j, s_j respectively denote the aging factor and the start time of job j. For the time-dependent model, Wu and Lee [14] studied the problem with a known maintenance period to minimize the makespan, and the problem was solved by the 0–1 integer programming method. With the assumption given by Wu and Lee [14], Ji et al. [15] considered the non-resumable model with the objective to minimize the makespan and total completion times. Both problems are proved to be NP-hard, and two pseudo-polynomial time algorithms were proposed to solve the problems. Furthermore, Lee and Wu [16] considered the multi-machine makespan minimization problem. For the resumable and non-resumable models, they proposed a heuristic algorithm for each model. Lee et al. [17] studied the multi-machine scheduling problem with the objective to minimize the total tardiness and earliness.

Zhan and Tang [18] considered another model of time-dependent aging effect, i.e., the actual processing time of job j is $p_j + \alpha s_j$, where p_j and α denote the processing load of job j and the common aging factor, respectively. The authors proposed polynomial time algorithms for the problems of minimizing the makespan and the total completion times. Yang and Yang [19] studied the scheduling models with multi-maintenance, and showed the problems are polynomially solvable. In addition, Yang [20] investigated the unrelated parallel-machine scheduling problem, and proposed a polynomial time algorithm for the special case in which the number of machines is known in advance. Lee et al. [21] studied the flowshop scheduling problem of minimizing the total tardiness cost. Yang et al. [22] considered the problems with slack due-date assignment.

Besides the type of time-dependent aging effect, Kuo and Yang [23] studied the position-dependent aging effect, i.e., if job j is processed in the rth position, then its actual processing time is $p_j r^{\alpha_j}$. For the model with at most one maintenance in the whole scheduling horizon, Yang et al. [24] proposed a polynomial time algorithm to solve the due-window assignment problem on a single machine. Zhao and Tang [25] studied the makespan minimization problem with multi-maintenance. Based on the assignment method, the authors proposed a polynomial time algorithm. Yang and Yang [26,27] studied several other problems with this type of aging effect, and derived a polynomial algorithm for each problem. Yin et al. [28] considered scheduling problems with due date determination. In addition, Yang and Yang [19] studied the problem to minimize the total completion times, where the actual processing of job j is $p_j + \alpha_j r$. Yang et al. [29] and Hsu et al. [30] further extended some results on position-dependent aging effect to mutli-machine scheduling problems.

The model with the time-dependent and position-dependent effects simultaneously was firstly studied by Yang [31]. In this model, the actual processing time of job j is defined as $(p_j - \beta s_r) r^{\alpha}$, where β is a common decreasing rate. For the scheduling with an optional maintenance, by the weight-matching technique [32], the author proposed a polynomial time algorithm for the problem to minimize the makespan, and the result was further extended to the total completion times minimization model.

This paper introduces a new type of aging effect, namely the machine aging effect, which is dominated by the processing speed of the machine. We assume the processing speed of the machine is a decreasing function of its continuously running time in the current uptime. Hence, in a common uptime, the later a job is scheduled, the more time is needed to process the job. The motivation for this type of aging effect stems from the requests serving process in the wireless sensor network. In a normal network, the time required for serving a request depends on the remaining energy of the sensor, and the available energy is decreasing during the serving process. Therefore, the actual serving time of a request increases with the continuous working time of the sensor in the current uptime. To counteract the decline of the serving ability, the maintenance activity (battery charging) may be performed on the sensor to maintain its serving ability, and the length of the downtime (i.e., the duration of a maintenance) depends on the length of the previous uptime.

Consequently, this paper firstly considers the single-machine scheduling problem with the combination of the machine aging effect and an optional maintenance operation. During the whole scheduling horizon, at most one maintenance activity is performed, and the duration of the optional maintenance depends on the length of the uptime before it. The objective is to find the position of the maintenance and the sequence of all jobs to minimize the makespan or the total completion times. We prove the two problems are NP-complete, and show that each of the two problems has an optimal processing sequence such that the process order of jobs before and after the maintenance are, respectively, subject to the *Shortest Processing Time first* (SPT) rule. In the rest of the paper, such a sequence is defined to be subject to **S-S** rule. Two *dynamic programming* (DP) algorithms are developed to solve the problem with an optimal sequence subject to S-S rule. Taking the problem with the objective to minimize the total completion times as an example, we introduce the details of the two DP algorithms. The analysis of the computational complexity of the algorithms indicates that the problem is polynomially solvable if the processing loads of all jobs are uniformly bounded.

The remainder of this paper is organized as follows. In Section 2, the problems under study are formulated. In Section 3, two problems with objective to minimize the makespan and the total completion times are proved to be NP-complete, and two DP algorithms are proposed in Section 4. Concluding remarks are given in Section 5.

Please cite this article as: M. Gu et al., Single-machine scheduling problems with machine aging effect and an optional maintenance activity, Applied Mathematical Modelling (2016), http://dx.doi.org/10.1016/j.apm.2016.01.038

2. Problem formulation and notation

The problems under study can be formulated as follows: There are n jobs, $\Omega = \{1, 2, \ldots, n\}$, to be processed on a single machine from time zero and job preemption is not allowed. The normal processing time of job j is p_j. The processing speed of the machine is defined as $v(t)$, where t denotes the time that the machine has been *continuously* processing jobs. Define $v(t)$ is a deceasing function satisfying $v(0) = 1$ and $v(t) \to 0 (t \to +\infty)$. Assume job j starts to be processed when the machine has already been continuously working for t_j time, then its actual processing time $\mathcal{P}_j$ satisfies

$$\int_{t_j}^{t_j+\mathcal{P}_j} v(t)dt = p_j, \tag{1}$$

i.e. given $v(t)$, the value of $\mathcal{P}_j$ depends on t_j and p_j. For notation convenience, we define a function int(x) such that

$$\int_0^{\text{int}(x)} v(t)dt = x.$$

Therefore, the actual processing time $\mathcal{P}_j = \text{int}\big(t_j + p_j\big) - \text{int}\big(t_j\big)$.

Due to the effect of machine aging, maintenance may be performed on the machine to improve its production efficiency. We assume that at most one maintenance operation is allowed throughout the scheduling horizon. The start time of the maintenance is not known in advance, and the maintenance operation can be carried out immediately after the completion of any job. The maintenance duration, denoted by $f(t)$, is a function of its start time t. After the maintenance, the machine will revert to its initial condition and start anew.

The objective is to find jointly the optimal maintenance position and the optimal job sequence to minimize the makespan and the total completion times. For a given schedule, denote by C_j the completion time of job j. Then the two objective functions can be presented as $\max_{1\le j\le n} C_j$ and $\sum_{j=1}^n C_j$.

3. Proof of NP-completeness

3.1. minimization of the makespan

In this subsection, we try to find the optimal maintenance position and the optimal job sequence so as to minimize the makespan. Following the three fields notation of Graham et al. [33], the problem is denoted by $1, v(t)|ma|C_{\max}$. Given a processing sequence, let $p_{[j]}$, $\mathcal{P}_{[j]}$ and $C_{[j]}$, respectively, denote the normal processing time, actual processing time and the completion time of the job processed in the jth position.

Assuming the maintenance operation is scheduled immediately after the job in the ith ($1 \le i \le n$) position, the completion times of jobs scheduled before the maintenance can be presented as follows.

$$C_{[j]} = \text{int}\big(S_{[1,j]}\big), 1 \le j \le i, \tag{2}$$

where $S_{[k,l]} = \sum_{m=k}^{l} p_{[m]} (k \le l)$, and the actual processing time

$$\mathcal{P}_{[j]} = \text{int}\big(S_{[1,j]}\big) - \text{int}\big(S_{[1,j-1]}\big),$$

where $S_{[1,0]} = 0$. Based on (2), it is easy to find the problem $1, v(t)||C_{\max}$ is trivial.

After the maintenance, the machine restores its initial condition and starts anew. Therefore, the actual processing time of jobs scheduled after the maintenance can be formulated as follows.

$$\mathcal{P}_{[j]} = \text{int}\big(S_{[i+1,j]}\big) - \text{int}\big(S_{[i+1,j-1]}\big), i+1 \le j \le n,$$

where $S_{[i+1,i]} = 0$, and the completion time is given by

$$C_{[j]} = C_{[i]} + f\big(C_{[i]}\big) + \sum_{k=i+1}^{j} \mathcal{P}_{[k]} \tag{3}$$

$$= \text{int}\big(S_{[1,i]}\big) + f\Big(\text{int}\big(S_{[1,i]}\big)\Big) + \text{int}\big(S_{[i+1,j]}\big). \tag{4}$$

The following part focuses on the computational complexity of the problem. In complexity theory, the principal way of proving a problem is NP-complete is to reduce a known NP-complete problem to the problem considered. The 2-Partition problem is known to be NP-complete [34] and in what follows we will describe a way of reducing (in polynomial time) instances of 2-Partition problem to instances of our problem, such that an instance of 2-Partition has a solution *if* and *only if* the corresponding instance of our scheduling problem has a solution.

2-Partition problem: Given a set of n position integers $x_1, x_2, \ldots, x_n$, is there a subset H of $\{1, 2, \ldots, n\}$ such that $\sum_{j\in H} x_j = b$, where $b = \frac{1}{2}\sum_{j=1}^n x_j$?

The computational complexity of the problem is stated in Theorem 2, before which we firstly introduce the following lemma.

Lemma 1. *Assuming* $v(t)$ *is a continuous and strictly decreasing function, then for any positive constant* c*, the function* $\mathrm{int}(x)+\mathrm{int}(2c-x)$ *is strictly increasing in the interval* $[c, 2c]$.

Proof. Given x_1, x_2 satisfying $c \le x_1 < x_2 \le 2c$, we have

$$\int_{\mathrm{int}(x_1)}^{\mathrm{int}(x_2)} v(t)dt = \int_0^{\mathrm{int}(x_2)} v(t)dt - \int_0^{\mathrm{int}(x_1)} v(t)dt = x_2 - x_1.$$

By the Mean Value Theorem for Integrals, there exists a constant ξ such that

$$\int_{\mathrm{int}(x_1)}^{\mathrm{int}(x_2)} v(t)dt = v(\xi)\Big[\mathrm{int}(x_2)-\mathrm{int}(x_1)\Big], \mathrm{int}(x_1) < \xi < \mathrm{int}(x_2).$$

Hence,

$$x_2 - x_1 = v(\xi)\Big[\mathrm{int}(x_2)-\mathrm{int}(x_1)\Big],$$

i.e.

$$\mathrm{int}(x_2)-\mathrm{int}(x_1) = \frac{x_2-x_1}{v(\xi)}. \tag{5}$$

With the same method, there exists a constant η such that

$$\mathrm{int}(2c-x_1)-\mathrm{int}(2c-x_2) = \frac{(2c-x_1)-(2c-x_2)}{v(\eta)}, \mathrm{int}(2c-x_2) < \eta < \mathrm{int}(2c-x_1),$$

i.e.

$$\mathrm{int}(2c-x_1)-\mathrm{int}(2c-x_2) = \frac{x_2-x_1}{v(\eta)}. \tag{6}$$

Based on (5) and (6), we have

$$\begin{aligned}&\Big[\mathrm{int}(x_2)+\mathrm{int}(2c-x_2)\Big]-\Big[\mathrm{int}(x_1)+\mathrm{int}(2c-x_1)\Big]\\ &= \Big[\mathrm{int}(x_2)-\mathrm{int}(x_1)\Big]-\Big[\mathrm{int}(2c-x_1)-\mathrm{int}(2c-x_2)\Big]\\ &= \frac{x_2-x_1}{v(\xi)}-\frac{x_2-x_1}{v(\eta)} = \frac{x_2-x_1}{v(\xi)v(\eta)}[v(\eta)-v(\xi)]>0,\end{aligned}$$

where the inequality holds because $v(t)$ is strictly decreasing and $\eta < \mathrm{int}(c) < \xi$. Therefore, the function $\mathrm{int}(x)+\mathrm{int}(2c-x)$ is strictly increasing in the interval $[c, 2c]$. □

Theorem 2. *The problem* $1, v(t)|ma|C_{\max}$ *is NP-complete.*

Proof. The proof is based on the following transformation by reduction from 2-Partition problem. Given an instance of 2-Partition problem, we construct the following instance of our scheduling problem.

Jobs: $1, 2, \ldots, n$;
Normal processing times: $p_j = x_j$, $1 \le j \le n$;
$v(t)$ is a *continuous and strictly* decreasing function;
$f(t) = 0$;
Makespan threshold value: $y = 2 \cdot \mathrm{int}(b)$.

Now we prove that the instance has a minimum objective value of y if and only if 2-Partition has a solution, which will then imply that our problem is NP-complete.

⇒ If there is a solution to 2-Partition instance, we show that there is a schedule to our problem with a makespan of no more than y. Given a solution to 2-Partition, i.e. there exists a subset H of $\{1, 2, \ldots, n\}$ such that $\sum_{j\in H} x_j = b$.

We construct a schedule for our problem, where jobs in H and $\Omega\backslash H$ are, respectively, processed before and after the maintenance. In this schedule, the times needed to finish the jobs in H and $\Omega\backslash H$ are both $\mathrm{int}(b)$. Since $f(t) = 0$, it is easy to see that the obtained makespan of our problem is $2 \cdot \mathrm{int}(b)(= y)$.

⇐ Conversely, assume there exists a schedule for the constructed instance of our problem with a makespan of no more than y. Let H_1, H_2 denote the jobs processed before and after the maintenance, respectively. Without lost of generality, assume $\sum_{j\in H_1} p_j \ge \sum_{j\in H_2} p_j$. Let $\delta = \sum_{j\in H_1} p_j$, then $b \le \delta \le 2b$. The makespan can be formulated as $\mathrm{int}(\delta)+\mathrm{int}(2b-\delta)$, and

$$\mathrm{int}(\delta)+\mathrm{int}(2b-\delta) \le y = 2\cdot \mathrm{int}(b). \tag{7}$$

On the other side, known from Lemma 1, $\mathrm{int}(x)+\mathrm{int}(2b-x)$ is strictly increasing in the interval $[b, 2b]$, then

$$\mathrm{int}(\delta)+\mathrm{int}(2b-\delta) \ge 2\cdot \mathrm{int}(b), \tag{8}$$

where the equality holds when $x=b$. Based on (7) and (8), we obtain

$$\mathrm{int}(\delta)+\mathrm{int}(2b-\delta)=2\cdot\mathrm{int}(b). \tag{9}$$

Hence $\delta=b$, i.e. $\sum_{j\in H_1}p_j=b$. Let $H=H_1$, then a solution to 2-Partition problem is obtained.

Combining the *if* part and *only if* part, we have proved the theorem. □

The expression (4) shows that the processing order of jobs before the maintenance has no effect on the makespan, nor does that of the jobs after the maintenance. Therefore, the following lemma holds trivially.

Lemma 3. *For the problem* $1, v(t)|ma|C_{\max}$, *there exits an optimal sequence subject to S-S rule.*

3.2. Minimization of total completion times

In this subsection, we try to find the optimal maintenance position and the optimal job sequence to minimize the total completion times. Following the three fields notation of Graham et al. [33], we denote the problem by $1, v(t)|am|\Sigma C_j$. Given a processing sequence λ, assume the maintenance operation is carried out immediately after the job in the ith position. Following (2) and (4), the total completion times is given by:

$$\pi(\lambda)=\sum_{j=1}^{i}C_{[j]}+\sum_{j=i+1}^{n}C_{[j]} \tag{10}$$

$$=\sum_{j=1}^{i}\mathrm{int}\left(S_{[1,j]}\right)+\sum_{j=i+1}^{n}\left[\mathrm{int}\left(S_{[1,i]}\right)+f\left(\mathrm{int}\left(S_{[1,i]}\right)\right)+\mathrm{int}\left(S_{[i+1,j]}\right)\right] \tag{11}$$

$$=\sum_{j=1}^{i}\mathrm{int}\left(S_{[1,j]}\right)+(n-i)\left[\mathrm{int}\left(S_{[1,i]}\right)+f\left(\mathrm{int}\left(S_{[1,i]}\right)\right)\right]+\sum_{j=i+1}^{n}\mathrm{int}\left(S_{[i+1,j]}\right). \tag{12}$$

The following theorem states the computational complexity of our problem, and proves the NP-completeness by reducing the 2-Partition problem into our problem in polynomial time.

Theorem 4. *The scheduling problem* $1, v(t)|ma|\Sigma C_j$ *is NP-complete.*

Proof. Given an instance of 2-Partition problem, the corresponding single-machine problem can be constructed as follows:

Jobs: $1,2,\ldots,n$;

Processing times: $p_j=2n\cdot x_j,\ j=1,2,\ldots,n$;

$v(t)=\frac{1}{1+t},\ f(t)=0$;

Threshold value of total completion times: $y=2n\cdot\exp(2nb)$, where $\exp(x)$ denotes the exponential function e^x. Next we will prove that the problem has a minimum objective value of no more than y if and only if 2-Partition problem has a solution, which then shows that our problem is NP-complete.

$\Rightarrow$ If there is a solution to 2-Partition instance, then there exists a solution to our problem with an objective value of no more than y. Denote by H the subset of $\{1,2,\ldots,n\}$ such that $\sum_{j\in H}x_j=b$. Arrange jobs in H and $\Omega\backslash H$ to be processed, respectively, before and after the maintenance. Denote the sequence by λ. Let i be the cardinality of H, i.e. $i=|H|$, then $p_{[1]}+\ldots+p_{[i]}=p_{[i+1]}+\ldots+p_{[n]}=2nb$, and

$$\begin{aligned}\mathrm{int}\left(S_{[1,i]}\right)&=\exp(2nb)-1,\\ \mathrm{int}\left(S_{[1,j]}\right)&\le\mathrm{int}\left(S_{[1,i]}\right),\ 1\le j\le i,\\ \mathrm{int}\left(S_{[i+1,n]}\right)&=\mathrm{int}\left(S_{[1,i]}\right)=\exp(2nb)-1,\\ \mathrm{int}\left(S_{[i+1,j]}\right)&\le\mathrm{int}\left(S_{[i+1,n]}\right),\ i+1\le j\le n.\end{aligned}$$

Noting $f(t)=0$, by (12), the total completion times is given by

$$\begin{aligned}\pi(\lambda)&=\sum_{j=1}^{i}\mathrm{int}\left(S_{[1,j]}\right)+(n-i)\mathrm{int}\left(S_{[1,i]}\right)+\sum_{j=i+1}^{n}\mathrm{int}\left(S_{[i+1,j]}\right)\\ &\le i\cdot\mathrm{int}\left(S_{[1,i]}\right)+(n-i)\mathrm{int}\left(S_{[1,i]}\right)+(n-i)\mathrm{int}\left(S_{[i+1,n]}\right)\\ &=(2n-i)\mathrm{int}\left(S_{[1,i]}\right)\\ &=(2n-i)\left[\exp(2nb)-1\right]\\ &\le 2n\cdot\exp(2nb)=y,\end{aligned}$$

i.e. we obtain a solution with an objective value of no more than y to our problem.

$\Leftarrow$ Assume there exists a solution for the constructed instance of our problem with an objective value of no more than y. We will prove that there is a solution to the 2-PARTITION problem. Let H_1 and H_2 be the two sets of jobs, respectively, processed before and after the maintenance, then $\sum_{j\in H_1} x_j = b$. We give the proof for the conclusion by contradiction.

If $\sum_{j\in H_1} x_j \neq b$, then $\sum_{j\in H_1} x_j \geq b+1$ or $\sum_{j\in H_2} x_j \geq b+1$. Letting $i = |H_1|$, we have $\text{int}\left(S_{[1,i]}\right) \geq \exp(2n(b+1)) - 1$ or $\text{int}\left(S_{[i+1,n]}\right) \geq \exp(2n(b+1)) - 1$. Therefore, the total completion times

$$\begin{aligned}
\pi(\lambda) &= \sum_{j=1}^{i} \text{int}\left(S_{[1,j]}\right) + (n-i)\text{int}\left(S_{[1,i]}\right) + \sum_{j=i+1}^{n} \text{int}\left(S_{[i+1,j]}\right) \\
&\geq \text{int}\left(S_{[1,i]}\right) + \text{int}\left(S_{[i+1,n]}\right) \\
&\geq \exp(2n(b+1)) - 1 \\
&= \exp(2n)\cdot\exp(2nb) - 1 \\
&> (2n+1)\cdot\exp(2nb) - 1 \\
&= 2n\cdot\exp(2nb) + \exp(2nb) - 1 \\
&> 2n\cdot\exp(2nb),
\end{aligned}$$

i.e. $\pi(\lambda) > 2n\cdot\exp(2nb) = y$, which contradicts the assumption. Therefore, $\sum_{j\in H_1} x_j = b$.

Based on the *if* part and *only if* part, we have proved the theorem.

□

Lemma 5. *For the problem* $1, v(t)|ma|\Sigma C_j$, *the optimal sequence is subject to S-S rule.*

Proof. Given a sequence λ, assume the maintenance activity is performed immediately after the job in the ith position.

Firstly, consider the jobs processed before the maintenance. Denote by $p^{\lambda}_{[j]}, p^{\lambda}_{[j+1]}$ $(1 \leq j \leq i-1)$ the normal processing times of the two jobs in the jth and $(j+1)$th positions in sequence λ. Let λ' be the new sequence by exchanging the processing order of the two jobs.

Without loss of generality, assuming $p^{\lambda}_{[j]} > p^{\lambda}_{[j+1]}$, then $S^{\lambda}_{[1,j]} > S^{\lambda'}_{[1,j]}$, and $\text{int}\left(S^{\lambda}_{[1,j]}\right) > \text{int}\left(S^{\lambda'}_{[1,j]}\right)$. The completion times of jobs in the same position in λ and λ' satisfies:

$$C^{\lambda}_{[j]} > C^{\lambda'}_{[j]} \text{and} C^{\lambda}_{[k]} = C^{\lambda'}_{[k]}, k \neq j.$$

Therefore,

$$\pi(\lambda) - \pi\left(\lambda'\right) = C^{\lambda}_{[j]} - C^{\lambda'}_{[j]} > 0.$$

i.e. exchanging the processing order of jobs in the jth and $(j+1)$th positions in λ will decrease the total completion times. Hence, the processing order of the jobs before the maintenance is subject to SPT rule.

With the same method, it can be proved that the processing of the jobs after the maintenance also follows SPT rule. Therefore, for the problem $1, v(t)|ma|\Sigma C_j$, the optimal sequence is subject to S-S rule. □

Based on Lemma 5, the following corollary holds immediately.

Corollary 6. *For the problem* $1, v(t)||\Sigma C_j$, *the optimal sequence is subject to SPT rule.*

In the next section, DP algorithms will be introduced to solve the problem with an optimal sequence subject to S-S rule.

4. Dynamic programming algorithms

For a NP-complete problem, although the mixed integer programming model can be used to obtain an optimal solution, constrains and variables increase drastically as the number of jobs increases. The theorems above indicate that there is no polynomial-time algorithm to solve the scheduling problems under study. Therefore, designing effective DP algorithms to achieve the optimal solution is of great interest.

Two DP algorithms will be proposed to solve the problem with an optimal sequence subject to S-S rule. For concision, we just focus on the problem with the objective to minimize the total completion times, and the makespan minimization problem can also be solved by developing similar DP algorithms.

Assume all jobs are indexed following the inequalities below:

$$p_1 \geq p_2 \geq \cdots \geq p_n.$$

Let $S_i = \{1, 2, \ldots, i\}$, and $\overline{S_i} = \Omega \backslash S_i$. Among the jobs in $\overline{S_i}$, define B_i be the set of jobs scheduled before the maintenance. Assume λ^* is the optimal sequence subject to S-S rule. Known from Lemma 5, all jobs in B_i are sequenced before S_i in sequence λ^*. On the other side, let $A_i = \overline{S_i} \backslash B_i$, i.e. among the jobs in $\overline{S_i}$, A_i denotes the set of jobs processed after the maintenance.

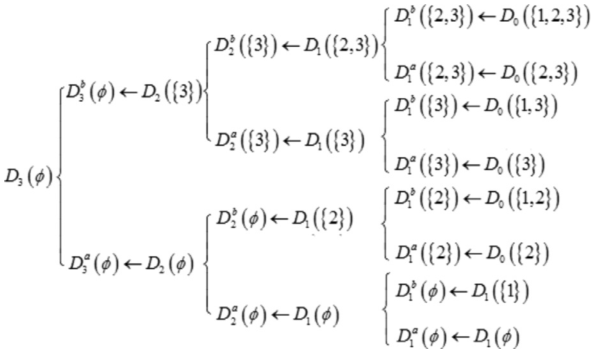

Fig. 1. Main procedure of Algorithm 1 on the Example 1.

Given i and B_i, define $D_i(B_i)$ as the contribution of all jobs in S_i to the objective function. In the optimal sequence λ^*, it is clear that if job i is sequenced before the maintenance, then i will be processed immediately after jobs in B_i, and

$$D_i(B_i) = D_i^b(B_i) = D_{i-1}(B_i \cup \{i\}) + \operatorname{int}\left(\sum_{j\in B_i\cup\{i\}} p_j\right) \tag{13}$$

$$+ |A_i| \cdot \left[f\left(\operatorname{int}\left(\sum_{j\in B_i\cup\{i\}} p_j\right)\right) - f\left(\operatorname{int}\left(\sum_{j\in B_i} p_j\right)\right)\right] \tag{14}$$

$$+ |A_i| \cdot \left[\operatorname{int}\left(\sum_{j\in B_i\cup\{i\}} p_j\right) - \operatorname{int}\left(\sum_{j\in B_i} p_j\right)\right], \tag{15}$$

where (13) denotes the completion time of job i, and the sum of (14) and (15) equals the total increased value of completion times of jobs in A_i. If job i is sequenced after the maintenance, then i will be processed immediately after jobs in A_i, and

$$D_i(B_i) = D_i^a(B_i) = D_{i-1}(B_i) + \operatorname{int}\left(\sum_{j\in B_i} p_j\right) + f\left(\operatorname{int}\left(\sum_{j\in B_i} p_j\right)\right) + \operatorname{int}\left(\sum_{j\in A_i\cup\{i\}} p_j\right), \tag{16}$$

where (16) denotes the completion time of job i. According to the definition of sequence λ^*, it is clear that in the optimal sequence λ^*, the job i must be scheduled such that

$$D_i(B_i) = \min\left\{D_i^b(B_i), D_i^a(B_i)\right\}. \tag{17}$$

The boundary conditions are

$$D_0(B_0) = 0, \text{for all} B_0 \in P(\Omega), \tag{18}$$

where $P(\Omega)$ denotes the power set of Ω. The algorithm proposed for the problem of minimizing the total completion times can be summarized as follows.

Algorithm 1.

Step 1. For $i = 1, 2, \ldots, n$, compute $D_i^b(B_i)$ and $D_i^a(B_i)$ for all $B_i \in P(\Omega \backslash S_i)$.
Step 2. Compute the minimal objective value $\pi(\lambda^*) = D_n(\phi)$, where ϕ denotes the empty set.
Step 3. By backtracking, construct the optimal sequence λ^*.

Next, a simple example is presented to illustrate the procedures of Algorithm 1.

Example 1. There are three jobs, i.e. $n = 3$, and $p_1 = 3, p_2 = 2, p_3 = 1$. The processing speed of the machine $v(t) = \frac{1}{1+0.5t}$, and the duration of the maintenance $f(t) = 1 + t$.

The main process of Algorithm 1 is described in Fig. 1, and the details of computation are given below.

Step 1: $i = 1, S_i = \{1\}, \Omega\backslash S_i = \{2, 3\}, B_i \in \{\phi, \{2\}, \{3\}, \{2, 3\}\}$.

$$D_1^b(\{2,3\}) = D_0(\{1,2,3\}) + \frac{1}{0.5}\left[e^{0.5\times(3+2+1)} - 1\right] = 2\left(e^3 - 1\right) \approx 38.1711.$$

Similarly,

$$D_1^a(\{2,3\}) \approx 21.8901, D_1^b(\{3\}) \approx 26.7049, D_1^a(\{3\}) \approx 25.9599,$$
$$D_1^b(\{2\}) \approx 36.2917, D_1^a(\{2\}) \approx 20.6512, D_1^b(\phi) \approx 34.8169, D_1^a(\phi) \approx 39.1711.$$

$i = 2, S_i = \{1, 2\}, \Omega \backslash S_i = \{3\}, B_i \in \{\phi, \{3\}\}.$

$$D_2^b(\{3\}) \approx 28.8535, D_2^a(\{3\}) \approx 32.9913, D_2^b(\phi) \approx 30.9609, D_2^a(\phi) \approx 42.7803.$$

$i = 3, S_i = \{1, 2, 3\}, \Omega \backslash S_i = \phi, B_i \in \{\phi\}.$

$$D_3^b(\phi) \approx 30.1510, D_3^a(\phi) \approx 33.2584.$$

Finally, we obtain

$$D_3(\phi) = \min\left\{D_3^b(\phi), D_3^a(\phi)\right\} \approx 30.1510.$$

Step 2. $\pi(\lambda^*) = D_3(\phi) \approx 30.1510.$
Step 3. By back tracking method, we obtain the optimal schedule (3, 2, *ma*, 1).

Known from the process description in Fig. 1, the Algorithm 1 needs $2^{n+1} - 1$ steps to compute all $D_n(\phi)$. Therefore, we have the following lemma.

Lemma 7. *For the problem* $1, v(t)|ma|\Sigma C_j$, *Algorithm 1 can find the optimal sequence* λ^* *in* $O\left(2^{n+1}\right)$ *time.*

Based on (13)–(16), we introduce another DP algorithm with computation time being polynomial if the normal processing times of all jobs are uniformly bounded.

Let $\mathcal{B}_i = \sum_{j \in B_i} p_j, b_i = |B_i|$. Given i, $\mathcal{B}_i$, and b_i, define $G_i(\mathcal{B}_i, b_i)$ as the contribution of all jobs in S_i to the objective value. If job i is sequenced before the maintenance, then

$$G_i(\mathcal{B}_i, b_i) = G_i^b(\mathcal{B}_i, b_i) = G_{i-1}(\mathcal{B}_i + p_i, b_i + 1) + \text{int}(\mathcal{B}_i + p_i) \tag{19}$$

$$+ (n - i - b_i)\Big[f(\text{int}(\mathcal{B}_i + p_i)) - f(\text{int}(\mathcal{B}_i))\Big] \tag{20}$$

$$+ (n - i - b_i)\Big[\text{int}(\mathcal{B}_i + p_i) - \text{int}(\mathcal{B}_i)\Big]. \tag{21}$$

Similarly, if job i is sequenced after the maintenance, then

$$G_i(\mathcal{B}_i, b_i) = G_i^a(\mathcal{B}_i, b_i) = G_{i-1}(\mathcal{B}_i, b_i) + \text{int}(\mathcal{B}_i) + f(\text{int}(\mathcal{B}_i)) + \text{int}\left(\sum_{j \in \overline{S_i}} p_j - \mathcal{B}_i + p_i\right). \tag{22}$$

According to the principle of optimality of dynamic programming, there are two positions for job i in the optimal sequence λ^*, and the job i must be scheduled such that

$$G_i(\mathcal{B}_i, b_i) = \min\left\{G_i^b(\mathcal{B}_i, b_i), G_i^a(\mathcal{B}_i, b_i)\right\}. \tag{23}$$

The boundary conditions are

$$G_0(\mathcal{B}_0, b_0) = 0, \text{for} \mathcal{B}_0 = 0, 1, \ldots, \sum_{j=1}^{n} p_j, \text{and} b_0 = 0, 1, \ldots, n. \tag{24}$$

The algorithm introduced above can be summarized as follows.

Algorithm 2.

Step 1. Based on (19)–(24), for $i = 1, 2, \ldots, n$, compute $G_i(\mathcal{B}_i, b_i)$ for all $\mathcal{B}_i = 0, 1, \ldots, \sum_{j \in \overline{S_i}} p_j$ and $b_i = 0, 1, \ldots, n - i$.
Step 2. Compute the minimal objective value $\pi(\lambda^*) = G_n(0, 0)$.
Step 3. By backward tracking, construct the optimal sequence λ^* .

The main process of Algorithm 2 applying to Example 1 is described in Fig. 2, and the details of computation are given below.

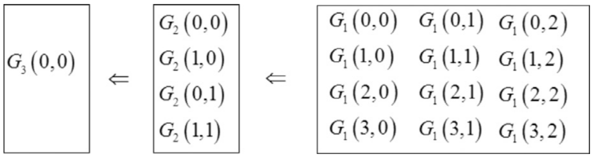

Fig. 2. Main procedure of Algorithm 1 on the Example 1.

Step 1: $i = 1, \mathcal{B}_i = 0, 1, 2, \ldots, \sum_{j\in\{2,3\}} p_j, b_i = 0, 1, 2,$

$$\begin{aligned}
G_1^b(0,0) &= G_0(3,1) + \text{int}(3) + 2\Big[f(\text{int}(3)) - f(\text{int}(0))\Big] + 2\Big[\text{int}(3) - \text{int}(0)\Big] \\
&= \frac{1}{0.5}\left(e^{0.5\times 3} - 1\right) + 2\left[\left(1 + \frac{1}{0.5}\left(e^{0.5\times 3} - 1\right)\right) - (1+0)\right] \\
&\quad + 2\left[\frac{1}{0.5}\left(e^{0.5\times 3} - 1\right) - 0\right] \\
&= 10\left(e^{1.5} - 1\right) \approx 34.8169, \\
G_1^a(0,0) &= G_0(0,0) + \text{int}(0) + f(\text{int}(0)) + \text{int}(6) \\
&= 1 + \frac{1}{0.5}\left(e^{0.5\times 6} - 1\right) \\
&= 1 + 2\left(e^{3} - 1\right) \approx 39.1711.
\end{aligned}$$

Therefore, we have

$$G_1(0,0) = \min\left\{G_1^b(0,0), G_1^a(0,0)\right\} \approx 34.8169.$$

Similarly,

$$\begin{aligned}
&G_1(1,0) \approx 25.9599, G_1(2,0) \approx 20.6512, G_1(3,0) \approx 21.8901, G_1(0,1) \approx 20.8901, \\
&G_1(1,1) \approx 25.9599, G_1(2,1) \approx 20.6512, G_1(3,1) \approx 21.8901, G_1(0,2) \approx 6.9634, \\
&G_1(1,2) \approx 12.7781, G_1(2,2) \approx 20.6512, G_1(3,2) \approx 21.8901.
\end{aligned}$$

$i = 2, \mathcal{B}_i = 0, \sum_{j\in\{3\}} p_j, b_i = 0, 1,$

$$G_2(0,0) \approx 30.9609, G_2(1,0) \approx 32.9913, G_2(0,1) \approx 24.0878, G_2(1,1) \approx 28.8535.$$

$i = 3, \mathcal{B}_i = 0, b_i = 0.$

$$G_3(0,0) \approx 30.1510.$$

Step 2. $\pi(\lambda^*) = G_3(0,0) \approx 30.1510.$
Step 3. By backtracking, we obtain the optimal schedule (3, 2, *ma*, 1).

Note that, for each i, the Algorithm 2 needs at most $n \cdot \sum_{j=1}^{n} p_j$ steps to compute all $G_i(\mathcal{B}_i, b_i)$. Therefore, we have the following lemma.

Lemma 8. *For the problem $1, v(t)|ma|\Sigma C_j$, Algorithm 2 can find the optimal sequence λ^* in $O(n^2\Phi)$ time, where $\Phi = \sum_{j=1}^{n} p_j$.*

Based on Lemma 8, the following corollary holds immediately.

Corollary 9. *For the problem $1, v(t)|ma|\Sigma C_j$, if the processing times of all jobs are uniformly bounded, then Algorithm 2 can find the optimal sequence λ^* in $O(n^3)$ time.*

5. Conclusions

This paper investigates single-machine scheduling problems with both aging effect, which is dominated by the processing speed of the machine, and deteriorating maintenance activities. We consider the model where the processing speed of the machine is a decreasing function of its uninterrupted running time. In addition, we assume the machine is subject to at most one maintenance activity during the scheduling horizon, and the maintenance duration is a general function of its start time. The objective is to find jointly the optimal location of the maintenance operation and the optimal job sequence to minimize the makespan and the total completion times. We prove the two problems under study are both NP-complete, and each of them has an optimal sequence subject to S-S rule. Taking the total completion times minimization problem as an example, we devise two DP algorithms to solve the problem with an optimal sequence subject to S-S rule. Furthermore, we analyze the computation complexity of the two algorithms, and show that the problem can be solved in polynomial time if the normal processing times of all jobs are uniformly bounded.

Please cite this article as: M. Gu et al., Single-machine scheduling problems with machine aging effect and an optional maintenance activity, Applied Mathematical Modelling (2016), http://dx.doi.org/10.1016/j.apm.2016.01.038

Further research may investigate the problem with model of multiple maintenance activities, in multi-machine settings, and different objective functions.

Acknowledgments

This work is supported by National Natural Science Foundation of China(Grant no. 11201282 and 61304209), Humanity and Social Science Youth Foundation of Ministry of Education (Grant no. 10YJCZH032), Innovation Program of Shanghai Municipal Education Commission (Grant no. 14YZ127).

References

[1] I. Adiri, J. Bruno, E. Frostig, A.H.G. Rinnooy Kan, Single machine flow-time scheduling with a single breakdown, Acta Informatica 26 (7) (1989) 679–696.
[2] G. Schmidt, Scheduling with limited machine availability, Eur. J. Oper. Res. 121 (2000) 1–15.
[3] Y. Ma, C. Chu, C. Zuo, A survey of scheduling with deterministic machine availability constraints, Comput. Indus. Eng. 58 (2010) 199–211.
[4] J.N.D. Gupta, S.K. Gupta, Single facility scheduling with nonlinear processing times, Comput. Indus. Eng. 14 (4) (1988) 387–393.
[5] A. Bachman, A. Janiak, Scheduling jobs with position-dependent processing times, J. Oper. Res. Soc. 55 (2004) 257–264.
[6] G. Mosheiov, Parallel machine scheduling with a learning effect, J. Oper. Res. Soc. 52 (2001) 1165–1169.
[7] S.H. Yang, J.B. Wang, Minimizing total weighted completion time in a two-machine flow shop scheduling under simple linear deterioration, Appl. Math. Comput. 217 (2011) 4819–4826.
[8] B. Aliaee, N.K. Womer, Scheduling with time dependent processing times: review and extensions, J. Oper. Res. Soc. 50 (1999) 711–720.
[9] D. Biskup, A state-of-the-art review on scheduling with learning effects, Eur. J. Oper. Res. 188 (2008) 315–329.
[10] T.C.E. Cheng, O. Ding, B.M.T. Lin, A concise survey of scheduling with time-dependent processing times, Eur. J. Oper. Res. 152 (2004) 1–13.
[11] A. Janiak, R. Rudek, Experience based approach to scheduling problems with learning effect, IEEE Trans. Syst. Man, Cybernetics-Part A 39 (2009) 344–357.
[12] K. Rustogi, V.A. Strusevich, Simple matching vs linear assignment in scheduling models with positional effects: acritical review, Eur. J. Oper. Res. 222 (2012) 393–407.
[13] S. Browne, U. Yechiali, Scheduling deteriorating jobs on a single processor, Oper. Res. 38 (1990) 495–498.
[14] C.C. Wu, W.C. Lee, Scheduling linear deteriorating jobs to minimize makespan with availability constraint on a single machine, Inf. Proc. Lett. 87 (2003) 89–93.
[15] M. Ji, Y. He, T.C.E. Cheng, Scheduling linear deteriorating jobs with an availability constraint on a single machine, Theoretical Comput. Sci. 362 (2006) 115–126.
[16] W.C. Lee, C.C. Wu, Multi-machine scheduling with deteriorating jobs and scheduled maintenance, Appl. Math. Model. 32 (2008) 362–373.
[17] H.T. Lee, D.L. Yang, S.J. Yang, Multi-machine scheduling with deterioration effects and maintenance activities for minimizing the total earliness and tardiness costs, Int. J. Adv. Manufacturing Technol. 66 (2013) 547–554.
[18] C.L. Zhao, H.Y. Tang, Single machine scheduling with past-sequence-dependent setup times and deteriorating jobs, Comput. Indus. Eng. 59 (2010) 663–666.
[19] S.J. Yang, D.L. Yang, Minimizing the total completion time in single-machine scheduling with aging/deteriorating effects and deteriorating maintenance activities, Comput. Math. Appl. 60 (2010) 2161–2169.
[20] S.J. Yang, Unrelated parallel-machine scheduling with deterioration effects and deteriorating mult-mainenance activities for minimizing the total completion time, Appl. Math. Model. 37 (2013) 2995–3005.
[21] W.C. Lee, W.C. Yeh, Y.H. Chung, Total tardiness minimization in permutation flowshop with deterioration consideration, Appl. Math. Model. 38 (13) (2014) 3081–3092.
[22] S.J. Yang, C.J. Hsu, D.L. Yang, Single-machine scheduling and slack due-date assignment with aging effect and deteriorating maintenance, Opt. Lett. 6 (2012) 1855–1873.
[23] W.H. Kuo, D.L. Yang, Minimizing the makespan in a single machine scheduling problem with the cyclic process of an aging effect, J. Oper. Res. Society 59 (2008) 416–420.
[24] S.J. Yang, D.L. Yang, T.C.E. Cheng, Single-machine due-window assignment and scheduling with job-dependent aging effects and deteriorating maintenance, Comput. Oper. Res. 37 (2010) 1510–1514.
[25] C.L. Zhao, H.Y. Tang, Single machine scheduling with general job-dependent aging effect and maintenance activities to minimize makespan, Appl. Math. Model. 34 (2010) 837–841.
[26] S.J. Yang, D.L. Yang, Minimizing the makespan on single-machine scheduling with aging effect and variable maintenance activities, Omega 38 (2010a) 528–533.
[27] S.J. Yang, D.L. Yang, Single-machine scheduling problems with aging/deteriorating effect under an optional maintenance activity consideration, Infor 48 (2010b) 171–179.
[28] Y. Yin, M. Liu, T.C.E. Cheng, C.C. Wu, S.R. Cheng, Four single-machine scheduling problems involving due date determination decisions, Inf. Sci. 251 (2013) 164–181.
[29] D.L. Yang, T.C.E. Cheng, S.J. Yang, C.J. Hsu, Unrelated parallel-machine scheduling with aging effects and multi-maintenance activities, Comput. Oper. Res. 39 (2012) 1458–1464.
[30] C.J. Hsu, M. Ji, J.Y. Guo, D.L. Yang, Unrelated parallel-machine scheduling problems with aging effects and deteriorating maintenance activities, Inf. Sci. 253 (2013) 163–169.
[31] S.J. Yang, Single-machine scheduling problems with both start-time dependent learning and position dependent aging effects under deteriorating maintenance consideration, Appl. Math. Comput. 217 (2010) 3321–3329.
[32] G.H. Hardy, J.E. Littlewood, G. Polya, Inequalities, Cambridge University Press, London, 1967.
[33] R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, Ann. Discrete Math. 5 (1979) 287–326.
[34] M.R. Garey, D.S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.

Please cite this article as: M. Gu et al., Single-machine scheduling problems with machine aging effect and an optional maintenance activity, Applied Mathematical Modelling (2016), http://dx.doi.org/10.1016/j.apm.2016.01.038