Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 19(11): 11756–11767.
DOI: 10.3934/mbe.2022547
Received: 12 June 2022
Revised: 26 July 2022
Accepted: 11 August 2022
Published: 16 August 2022

Research article

A maintenance activity scheduling with time-and-position dependent deteriorating effects

Weiguo Liu*, **Xuyin Wang** , **Lu Li and Peizhen Zhao**

Business School, Northwest Normal University, Lanzhou 730070, China

* **Correspondence:** Email: winterplum.home@163.com.

Abstract: We deal with a single-machine scheduling problem with an optional maintenance activity (denoted by *ma*), where the actual processing time of a job is a function of its starting time and position. The optional *ma* means that the machine will perform a *ma*, after *ma* is completed, the machine will return to the initial state. The objective is to determine an optimal job sequence and the location of the maintenance activity such that makespan is to be minimized. Based on some properties of an optimal sequence, we introduce a polynomial time algorithm to solve the problem, and the time complexity is $O(n^4)$, where n is the number of jobs.

Keywords: scheduling; deterioration effect; maintenance activity; single-machine

1. Introduction

In actual industrial production, the processing times of jobs have deterioration (aging) effects, i.e., the later of a job starts, the longer it takes to process it (see Wang and Wang [1]). Wang and Wang [2] considered the single-machine makespan minimization problem with time dependent processing times and group technology. Under ready times of the jobs, they proved that the problem can be solved in polynomial time. Wang and Wang [3] investigated the single-machine weighted sum of the hth power of waiting times problems with simple linear deterioration and precedence constraints. Under some precedence constraints, they proved that these problems remain polynomially solvable. Cheng et al. [4] studied single-machine problems with an accelerating deterioration effect. They proved that some regular and non-regular objective functions can be solved in polynomial time. Yin et al. [5] addressed some two-agent scheduling problems with deterioration effects on a single-machine. Zhang et al. [6] considered single-machine problems with time-dependent processing times. Under the common and slack due window assignments, they proved that two non-regular objective function minimizations can be solved in polynomial time. Liu et al. [7] considered single-machine group scheduling with deteriorating jobs. For the makespan minimization with ready times, they proposed branch-and-bound algorithm and heuristic algorithm. Wang and Liang [8] studied single-machine group scheduling with deterioration effects and resource allocation. For the makespan minimization under limited resource availability constraint, they proposed branch-and-bound and heuristic algorithms. Gawiejnowicz [9] reviewed scheduling problems with deteriorations effects (time-dependent processing times). In addition, with the deterioration (aging) effects, the machine can perform the maintenance activities and increase work efficiency, so, the scheduling problems with maintenance activities have also been considered (see Ma et al. [10] and Wang and Wei [11]). Hsu et al. [12] considered unrelated parallel-machine problem with rate-modifying activities. For the total completion time minimization, they proposed a more efficient algorithm. Ji et al. [13] studied single-machine slack due date assignment problem with job-dependent aging effects. Under a deteriorating maintenance activity, that proved that a non-regular objective minimization can be solved in polynomial time. Rustogi and Strusevich [14, 15] considered single-machine scheduling problems with rate modifying activities. Liu et al. [16] studied single-machine multiple common due-date assignments scheduling with deterioration effects. Under an maintenance activity, they proved that the linear weighted sum of earliness, tardiness, and the due dates minimization can be solved in polynomial time. Xiong et al. [17] considered the single-machine common due date assignment problem potential machine disruption. Zhu et al. [18] investigated multitasking scheduling problems with multiple rate-modifying activities. For the single-criterion and multi-criteria minimizations, they proposed optimal algorithms.

Recently, Zhang et al. [19] studied machine scheduling problems with deteriorating effects. Under the deteriorating rate-modifying activities, they proved that some objective function minimizations can be solved in polynomial time. Wang and Li [20] studied the unrelated parallel processors problem with deterioration effects and deteriorating maintenance activities. For some regular objective function minimizations, they proved that the problem can be solved in polynomial time. Zhang et al. [21] considered parallel machines scheduling problems with linear deteriorating jobs and maintenance activities. Under the resumable and non-resumable cases, the goal is to minimize the expected sum of completion times. They presented the pseudo-polynomial time algorithms to solve the problems. Zhang et al. [22] addressed single-machine scheduling problems with multiple maintenance activities and position-and-resource-dependent processing times. For some regular and non-regular objective function minimizations, they proposed polynomial time and pseudo-polynomial time algorithms. Sun and Geng [23] investigated the single-machine maintenance activity scheduling with deteriorating effects. The goal is to minimize the makespan and total completion time, they showed that the problem can be solved in polynomial time, respectively. Wang et al. [24] examined the single-machine common due-window assignment problem with a maintenance activity. Under constant and time-dependent processing times, they proved that a non-regular general earliness and tardiness minimization can be solved in polynomial time. Jia et al. [25] considered the single-machine scheduling problem with deterioration effects. Under the deterioration maintenance activity and slack due-window assignment, they showed that a non-regular objective function minimization can be solved in polynomial time. He et al. [26] discussed unrelated parallel processor scheduling with deterioration effects and maintenance activities. The objective is to minimize the total completion time and total machine load, they showed that these problems remain polynomially solvable.

The phenomena of deteriorating jobs and machine maintenance activity occurring simultaneously can be found in real-life situations. For example, consider a set of tasks (jobs) available for processing by a human operator. Perhaps the most distinguishing factor that differentiates human operators from machines within the context of task-sequencing is the notion of fatigue and its effect on task processing. More specifically, the rate at which a human operator performs a given task is known to be a decreasing function of the operator‘s level of fatigue, which manifests as the task processing time taking longer than expected. Another distinguishing characteristic is that human operators regularly engage in rest breaks during work shifts, which allows them to recover and mitigate some of the negative effects of fatigue (Sun and Geng [23]; Eilon [27]; Lodree and Geiger [28]). In this paper, we extend the results of Sun and Geng [23], by studying a more general processing times that includes the one given in Sun and Geng [23] as a special case. For the makespan minimization, we prove that the problem is polynomial-time solvable.

The organization of this article is as follows. Section 2 gives a description of the problem. Section 3 presents a polynomial-time solution for the problem. Conclusions are presented in Section 4.

2. Problem description

There are $\ddot{n}$ independent jobs $T_1, T_2, \ldots, T_n$ to be processed on a single-machine, and before processing, the machine needs $\tilde{t}$ ($\tilde{t} > 0$) preparation time. Due to the time-and-position dependent deteriorating effects (denoted by *tpdde*), the machine's production efficiency is reduced, the machine will perform a maintenance activity (denoted by *ma*), and when the repair is completed, the machine will return to the initial setting and the *tpdde* will start again.

It is assumed that the machine performs a *ma* after the kth job is completed, and the maintenance time is D. In the case of *tpdde*, the actual processing time of job T_j if it is scheduled in rth position in a sequence is given by

$$p_j = \begin{cases} (a_j + b_j s_j) r^c, & \text{if } r \le k, \\ (\theta_j a_j + b_j(s_j - C_{[k]} - D))(r-k)^c, & \text{if } r > k, \end{cases} \tag{1}$$

where a_j (resp. b_j) is the normal processing time (resp. Deteriorating rate) of T_j ($j = 1, 2, \ldots, n$), s_j (resp. θ_j ($\theta_j > 0$)) is the starting processing time (resp. Maintenance rate) of T_j, c ($c \ge 0$) is the common aging rate for all the jobs, and $[k]$ is some job scheduled in kth position. Ji et al. [29] studied the following (i.e., simple linear deterioration and aging) model $p_j = \begin{cases} b_j s_j r^c, & \text{if } r \le k, \\ b_j(s_j - C_{[k]} - D)(r-k)^c, & \text{if } r > k. \end{cases}$ Ji et al. [29] proved that the makespan minimization under simple linear deterioration and aging model is NP-hard in the strong sense. Sun and Geng [23] considered the following (i.e., linear deterioration) model $p_j = \begin{cases} a_j + b s_j, & \text{if } r \le k, \\ \theta a_j + b(s_j - C_{[k]} - D), & \text{if } r > k, \end{cases}$ where b (resp. $0 < \theta \le 1$) is the common deterioration rate (resp. Maintenance rate) for all the jobs. Sun and Geng [23] proved that the makespan and total completion time minimizations can be solved in polynomial time, respectively.

In this paper, we mainly concentrate on the following model:

$$p_j = \begin{cases} (a_j + b s_j) r^c, & \text{if } r \le k, \\ (\theta_j a_j + b(s_j - C_{[k]} - D))(r-k)^c, & \text{if } r > k, \end{cases} \tag{2}$$

Let C_j denote the completion time of job T_j, the objective is to determine the location of the maintenance activity and the job sequence φ such that the makespan $C_{\max} = \max\{C_1, C_2, \ldots, C_n\}$ is to be minimized. From Gawiejnowicz [9], this problem can be written as:

$$1|ma, tpdde|C_{\max} \tag{3}$$

3. Main results

The machine is repaired after the kth job is processed, the machine can be repaired in any position from 1 to $n-1$, and the maintenance time is D, hence, from Figure 1, we have

$$C_{\max} = \tilde{t} + \sum_{j=1}^{k} p_{[j]} + D + \tilde{t} + \sum_{j=k+1}^{n} p_{[j]} = D + 2\tilde{t} + \sum_{j=1}^{k} p_{[j]} + \sum_{j=k+1}^{n} p_{[j]} \tag{4}$$

$\tilde{t}$	$J_{[1]}$	$J_{[2]}$	$\cdots$	$J_{[k]}$	D	$\tilde{t}$	$J_{[k+1]}$	$J_{[k+2]}$	$\cdots$	$J_{[n]}$

Figure 1. A sample of $1|ma, tpdde|C_{\max}$.

According to the location of ma, we are divided into the following three cases, that is, case i: $k = 0$, case ii: $0 < k < n$, and case iii: $k = n$.

3.1. *Case i ($k = 0$)*

If $k = 0$, the actual machining time $p_{[j]}$ of the job $T_{[j]}$ is

$$p_{[j]} = \left(\theta_{[j]}a_{[j]} + b(s_{[j]} - D)\right) j^c, j = 1, 2, \ldots, n. \tag{5}$$

If $j = 1$, the machine needs $\tilde{t}$ time to prepare and maintenance time D,
$s_{[1]} = \tilde{t} + D$, $p_{[1]} = (\theta_{[1]}a_{[1]} + b(s_{[1]} - D))\, 1^c = \theta_{[1]}a_{[1]} + b\tilde{t}$.
If $j = 2$, $s_{[2]} = C_{[1]} = p_{[1]} + s_{[1]} = \theta_{[1]}a_{[1]} + (1 + b)\tilde{t} + D$,

$$\begin{aligned} p_{[2]} &= (\theta_{[2]}a_{[2]} + b(s_{[2]} - D))\, 2^c \\ &= \theta_{[2]}a_{[2]}2^c + b2^c(\theta_{[1]}a_{[1]} + (1 + b)\tilde{t}) \\ &= \theta_{[2]}a_{[2]}2^c + b2^c\theta_{[1]}a_{[1]} + b2^c(1 + b)\tilde{t}. \end{aligned}$$

If $j = 3$, $s_{[3]} = C_{[2]} = p_{[2]} + s_{[2]} = \theta_{[2]}a_{[2]}2^c + \theta_{[1]}a_{[1]}(1 + b2^c) + \tilde{t}(1 + b)(1 + b2^c) + D$,
$p_{[3]} = (\theta_{[3]}a_{[3]} + b(s_{[3]} - D))\, 3^c = \theta_{[3]}a_{[3]}3^c + b3^c(\theta_{[2]}a_{[2]}2^c + \theta_{[1]}a_{[1]}(1 + b2^c) + \tilde{t}(1 + b)(1 + b2^c))$.
......
If $j = i$,
$s_{[i]} = C_{[i-1]} = \sum_{h=1}^{i-1} \theta_{[h]}a_{[h]}h^c \prod_{l=h+1}^{i-1}(1 + bl^c) + \tilde{t}\prod_{l=1}^{i-1}(1 + bl^c) + D$,
$p_{[i]} = \theta_{[i]}a_{[i]}i^c + bi^c\left(\sum_{h=1}^{i-1} a_{[h]}h^c \prod_{l=h+1}^{i-1}(1 + bl^c) + \tilde{t}\prod_{l=1}^{i-1}(1 + bl^c)\right)$.
......
If $j = n$,
$s_{[n]} = C_{[n-1]} = \sum_{h=1}^{n-1} \theta_{[h]}a_{[h]}h^c \prod_{l=h+1}^{n-1}(1 + bl^c) + \tilde{t}\prod_{l=1}^{n-1}(1 + bl^c) + D$,
$p_{[n]} = \theta_{[n]}a_{[n]}n^c + bn^c\left(\sum_{h=1}^{n-1} a_{[h]}h^c \prod_{l=h+1}^{n-1}(1 + bl^c) + \tilde{t}\prod_{l=1}^{n-1}(1 + bl^c)\right)$.
By the simple algebraic calculation, we have

$$C_{\max} = C_{[n]}$$

$$
\begin{aligned}
&= s_{[n]} + p_{[n]} \\
&= \sum_{h=1}^{n-1} \theta_{[h]} a_{[h]} h^c \prod_{l=h+1}^{n-1} (1 + bl^c) + \tilde{t} \prod_{l=1}^{n-1} (1 + bl^c) + D \\
&\quad + \theta_{[n]} a_{[n]} i^c + bn^c \left(\sum_{h=1}^{n-1} a_{[h]} h^c \prod_{l=h+1}^{n-1} (1 + bl^c) + \tilde{t} \prod_{l=1}^{n-1} (1 + bl^c) \right) \\
&= \left(1 + b2^c + b3^c(1 + b2^c) + b4^c(1 + b2^c)(1 + b3^c) + \cdots + bn^c \prod_{h=2}^{n-1} (1 + bh^c) \right) \theta_{[1]} a_{[1]} \\
&\quad + \left(2^c + b3^c 2^c + b4^c 2^c (1 + b3^c) + \cdots + bn^c 2^c \prod_{h=3}^{n-1} (1 + bh^c) \right) \theta_{[2]} a_{[2]} \\
&\quad + \ldots + ((n-1)^c + bn^c (n-1)^c) \theta_{[n-1]} a_{[n-1]} + n^c \theta_{[n]} a_{[n]} \\
&\quad + \left(b + b2^c(1 + b) + b3^c(1 + b)(1 + b2^c) + \ldots + bn^c \prod_{l=1}^{n-1} (1 + bl^c) \right) \tilde{t} + D \\
&= \sum_{j=1}^{n} \Delta_j \theta_{[j]} a_{[j]} + E, \qquad (6)
\end{aligned}
$$

where

$$
\begin{aligned}
&\Delta_1 = 1 + b2^c + b3^c(1 + b2^c) + b4^c(1 + b2^c)(1 + b3^c) + \cdots + bn^c \prod_{h=2}^{n-1} (1 + bh^c) \\
&\Delta_2 = 2^c + b3^c 2^c + b4^c 2^c (1 + b3^c) + \cdots + bn^c 2^c \prod_{h=3}^{n-1} (1 + bh^c) \\
&\ldots\ldots \\
&\Delta_{n-1} = (n-1)^c + bn^c (n-1)^c \\
&\Delta_n = n^c,
\end{aligned}
\qquad (7)
$$

and

$$
E = D + \left(b + b2^c(1 + b) + b3^c(1 + b)(1 + b2^c) + \ldots + bn^c \prod_{l=1}^{n-1} (1 + bl^c) \right) \tilde{t}. \qquad (8)
$$

3.2. Case ii ($0 < k < n$)

If $0 < k < n$, the actual processing time of job T_j is:

$$
p_{[j]} = \begin{cases} (a_{[j]} + bs_{[j]}) j^c, & \text{if } j \le k \\ (\theta_{[j]} a_{[j]} + b(s_{[j]} - C_{[k]} - D))(j - k)^c, & \text{if } j > k \end{cases} \qquad (9)
$$

If $j = 1, 2, \ldots, k$, we have $s_{[1]} = \tilde{t}$, $p_{[1]} = (a_{[1]} + bs_{[1]}) 1^c = a_{[1]} + b\tilde{t}$,
$s_{[2]} = C_{[1]} = p_{[1]} + s_{[1]} = a_{[1]} + (1 + b)\tilde{t}$,
$p_{[2]} = (a_{[2]} + bs_{[2]}) 2^c = a_{[2]} 2^c + b2^c (a_{[1]} 1^c + (1 + b)\tilde{t})$,
$s_{[3]} = C_{[2]} = p_{[2]} + s_{[2]} = a_{[2]} 2^c + a_{[1]} 1^c (1 + b2^c) + (1 + b)(1 + b2^c)\tilde{t}$,
$p_{[3]} = (a_{[3]} + bs_{[3]}) 3^c = a_{[3]} 3^c + b3^c (a_{[2]} 2^c + a_{[1]} (1 + b2^c) + (1 + b)(1 + b2^c)\tilde{t})$,
......
$s_{[k]} = C_{[k-1]} = \sum_{h=1}^{k-1} a_{[h]} h^c \prod_{l=h+1}^{k-1} (1 + bl^c) + \tilde{t} \prod_{l=1}^{k-1} (1 + bl^c)$,
$p_{[k]} = (a_{[k]} + bs_{[k-1]}) i^c = a_{[k]} k^c + bk^c \left(\sum_{h=1}^{k-1} a_{[h]} h^c \prod_{l=h+1}^{k-1} (1 + bl^c) + \tilde{t} \prod_{l=1}^{k-1} (1 + bl^c) \right)$,

$C_{[k]}=\sum_{h=1}^{k}a_{[h]}h^c\prod_{l=h+1}^{k}(1+bl^c)+\tilde{t}\prod_{l=1}^{k}(1+bl^c)$.
If $j=k+1$, the machine need to be repaired after the job $T_{[k]}$ is completed, we have

$$s_{[k+1]}=C_{[k]}+D+\tilde{t},$$

$$p_{[k+1]}=(\theta_{[k+1]}a_{[k+1]}+b(s_{[k+1]}-C_{[k]}-D))1^c=\theta_{[k+1]}a_{[k+1]}+b\tilde{t}.$$

If $j=k+2$, $j=k+3,\ldots,j=n$, we have
$s_{[k+2]}=C_{[k+1]}=\theta_{[k+1]}a_{[k+1]}+(1+b)\tilde{t}+C_{[k]}+D$,
$p_{[k+2]}=(\theta_{[k+2]}a_{[k+2]}+b(s_{[k+2]}-C_{[k]}-D))2^c=\theta_{[k+2]}a_{[k+2]}2^c+b2^c\theta_{[k+1]}a_{[k+1]}+b2^c(1+b)\tilde{t}$.
$s_{[k+3]}=C_{[k+2]}=\theta_{[k+2]}a_{[k+2]}2^c+\theta_{[k+1]}a_{[k+1]}(1+b2^c)+(1+b)(1+b2^c)\tilde{t}+C_{[k]}+D$,

$$\begin{aligned}p_{[k+3]}&=(\theta_{[k+3]}a_{[k+3]}+b(s_{[k+3]}-C_{[k]}-d))3^c\\&=\theta_{[k+3]}a_{[k+3]}3^c+b3^c(\theta_{[k+2]}a_{[k+2]}2^c+\theta_{[k+1]}a_{[k+1]}(1+b2^c)+(1+b)(1+b2^c)\tilde{t})\end{aligned}$$

......
$s_{[n]}=C_{[n-1]}=\sum_{h=1}^{n-k-1}\theta_{[k+h]}a_{[k+h]}h^c\prod_{l=h+1}^{n-k-1}(1+bl^c)+\tilde{t}\prod_{l=1}^{n-k-1}(1+bl^c)+C_{[k]}+D$,

$$p_{[n]}=\theta_{[n]}a_{[n]}(n-k)^c+b(n-k)^c\left(\sum_{h=1}^{n-k-1}\theta_{[k+h]}a_{[k+h]}h^c\prod_{l=h+1}^{n-k-1}(1+bl^c)+\tilde{t}\prod_{l=1}^{n-k-1}(1+bl^c)\right).$$

Hence,

$$\begin{aligned}C_{\max}&=s_{[n]}+p_{[n]}\\&=\sum_{h=1}^{n-k-1}\theta_{[k+h]}a_{[k+h]}h^c\prod_{l=h+1}^{n-k-1}(1+bl^c)+\tilde{t}\prod_{l=1}^{n-k-1}(1+bl^c)\\&\quad+\sum_{h=1}^{k}a_{[h]}h^c\prod_{l=h+1}^{k}(1+bl^c)+\tilde{t}\prod_{l=1}^{k}(1+bl^c)+D\\&\quad+\theta_{[n]}a_{[n]}(n-k)^c+b(n-k)^c\left(\sum_{h=1}^{n-k-1}\theta_{[k+h]}a_{[k+h]}h^c\prod_{l=h+1}^{n-k-1}(1+bl^c)+\tilde{t}\prod_{l=1}^{n-k-1}(1+bl^c)\right)\\&=\left(1+b2^c+b3^c(1+b2^c)+b4^c(1+b2^c)(1+b3^c)+\cdots+bk^c\prod_{h=2}^{k-1}(1+bh^c)\right)a_{[1]}\\&\quad+\left(2^c+b3^c2^c+b4^c2^c(1+b3^c)+\cdots+bk^c2^c\prod_{h=3}^{k-1}(1+bh^c)\right)a_{[2]}\\&\quad+\ldots+((k-1)^c+bk^c(k-1)^c)a_{[k-1]}+k^ca_{[k]}\\&\quad+\left(\begin{array}{l}+b2^c+b3^c(1+b2^c)+b4^c(1+b2^c)(1+b3^c)\\+\cdots+b(n-k)^c\prod_{h=2}^{n-k-1}(1+bh^c)\end{array}\right)\theta_{[k+1]}a_{[k+1]}\\&\quad+\left(2^c+b3^c2^c+b4^c2^c(1+b3^c)+\cdots+b(n-k)^c2^c\prod_{h=3}^{n-k-1}(1+bh^c)\right)\theta_{[k+2]}a_{[k+2]}\\&\quad+\ldots+((n-k-1)^c+b(n-k-1)^c(n-k)^c\theta_{[n-1]}a_{[n-1]}+(n-k)^c\theta_{[n]}a_{[n]}\end{aligned}$$

$$
\begin{aligned}
& +\begin{pmatrix} b + b2^c(1+b) + b3^c(1+b)(1+b2^c) + ... + bk^c \prod_{l=1}^{k-1}(1+bl^c) \\ +b + b2^c(1+b) + ... + b(n-k)^c \prod_{l=1}^{n-k-1}(1+bl^c) \end{pmatrix}\tilde{t} + D \\
= & \sum_{j=1}^{k} \Delta_j a_{[j]} + \sum_{j=k+1}^{n} \Delta_j \theta_{[j]} a_{[j]} + E \qquad (10)
\end{aligned}
$$

where,

$$
\begin{aligned}
&\Delta_1 = 1 + b2^c + b3^c(1+b2^c) + b4^c(1+b2^c)(1+b3^c) + \cdots + bk^c \prod_{h=2}^{k-1}(1+bh^c) \\
&\Delta_2 = 2^c + b3^c2^c + b4^c2^c(1+b3^c) + \cdots + bk^c2^c \prod_{h=3}^{k-1}(1+bh^c) \\
&\ldots\ldots \\
&\Delta_{k-1} = (k-1)^c + bk^c(k-1)^c \\
&\Delta_k = k^c \\
&\Delta_{k+1} = 1 + b2^c + b3^c(1+b2^c) + b4^c(1+b2^c)(1+b3^c) \\
&\qquad + \cdots + b(n-k)^c \prod_{h=2}^{n-k-1}(1+bh^c) \\
&\Delta_{k+2} = 2^c + b3^c2^c + b4^c2^c(1+b3^c) + \cdots + b(n-k)^c2^c \prod_{h=3}^{n-k-1}(1+bh^c) \\
&\ldots\ldots \qquad (11) \\
&\Delta_{n-1} = (n-k-1)^c + b(n-k-1)^c(n-k)^c \\
&\Delta_n = (n-k)^c,
\end{aligned}
$$

and

$$
E = D + \tilde{t}\begin{pmatrix} b + b2^c(1+b) + b3^c(1+b)(1+b2^c) + ... + bk^c \prod_{l=1}^{k-1}(1+bl^c) \\ +b + b2^c(1+b) + ... + b(n-k)^c \prod_{l=1}^{n-k-1}(1+bl^c) \end{pmatrix}. \qquad (12)
$$

3.3. *Case iii ($k = n$)*

Similarly, if $k = n$, the actual processing time of $T_{[j]}$ is

$$
p_{[j]} = \left(a_{[j]} + bs_{[j]}\right) j^c, \; j = 1, 2, \ldots, n, \qquad (13)
$$

we have

$$
\begin{aligned}
C_{\max} = & (a_{[1]} + b\tilde{t}) + (a_{[2]}2^c + b2^c a_{[1]} + b2^c(1+b)\tilde{t}) \\
& +... + \left(a_{[n]}n^c + bn^c \left(\sum_{h=1}^{n-1} a_{[h]}h^c \prod_{l=h+1}^{n-1}(1+bl^c) + \tilde{t}\prod_{l=1}^{n-1}(1+bl^c)\right)\right) + D \\
= & \left(1 + b2^c + b3^c(1+b2^c) + b4^c(1+b2^c)(1+b3^c) + \cdots + bn^c \prod_{h=2}^{n-1}(1+bh^c)\right) a_{[1]} \\
& + \left(2^c + b3^c2^c + b4^c2^c(1+b3^c) + \cdots + bn^c2^c \prod_{h=3}^{n-1}(1+bh^c)\right) a_{[2]} \\
& +... + ((n-1)^c + bn^c(n-1)^c)a_{[n-1]} + n^c a_{[n]} \\
& + \left(b + b2^c(1+b) + b3^c(1+b)(1+b2^c) + ... + bn^c \prod_{l=1}^{n-1}(1+bl^c) \right)\tilde{t} + D \\
= & \sum_{j=1}^{n} \Delta_j a_{[j]} + E \qquad (14)
\end{aligned}
$$

where

$$\begin{array}{l}\Delta_1 = 1 + b2^c + b3^c(1 + b2^c) + b4^c(1 + b2^c)(1 + b3^c) + \cdots + bn^c \prod_{h=2}^{n-1}(1 + bh^c)\\ \Delta_2 = 2^c + b3^c2^c + b4^c2^c(1 + b3^c) + \cdots + bn^c2^c \prod_{h=3}^{n-1}(1 + bh^c)\\ \ldots\ldots\\ \Delta_{n-1} = (n-1)^c + bn^c(n-1)^c\\ \Delta_n = n^c,\end{array} \tag{15}$$

and

$$E = D + \left(b + b2^c(1 + b) + b3^c(1 + b)(1 + b2^c) + \ldots + bn^c \prod_{l=1}^{n-1}(1 + bl^c)\right)\tilde{t}. \tag{16}$$

3.4. Optimal solution

Lemma 1. *(Hardy et al. [30]) $\sum_{i=j}^{n} \mu_j \nu_j$ get the minimum, when sequence μ_j and sequence ν_j is arranged in opposite monotonous order.*

From Eqs (8), (12) and (16), we have that E is a constant.

For case i (i.e., $k = 0$), the problem $1|ma, tpdde|C_{\max}$ can be easily solved by Lemma 1 in $O(n \log n)$ time, i.e., $\mu_j = \Delta_j$ (see Eqs (6) and (7)), $\nu_j = \theta_j a_j$.

For case iii (i.e., $k = n$), the problem $1|ma, tpdde|C_{\max}$ can be easily solved by Lemma 1 in $O(n \log n)$ time, i.e., $\mu_j = \Delta_j$ (see Eqs (14) and (15)), $\nu_j = a_j$.

For case ii (i.e., $0 < k < n$), let $Z_{j,r} = 1$ if job T_j is scheduled in rth position, and $Z_{j,r} = 0$, otherwise. Form Eq (10), the optimal solution of the problem $1|ma, tpdde|C_{\max}$ can be solved by the following assignment problem:

$$\text{Min}\left(\sum_{j=1}^{n}\sum_{r=1}^{k} \Delta_r a_j Z_{j,r} + \sum_{j=1}^{n}\sum_{r=k+1}^{n} \Delta_r \theta_j a_j Z_{j,r}\right) \tag{17}$$

Subject to

$$\sum_{j=1}^{n} Z_{j,r} = 1, \quad r = 1, 2, \ldots, n, \tag{18}$$

$$\sum_{r=1}^{n} Z_{j,r} = 1, \quad j = 1, 2, \ldots, n, \tag{19}$$

$$Z_{j,r} = 0 \text{ or } 1, \quad j, r = 1, 2, \ldots, n, \tag{20}$$

where Δ_r $(r = 1, 2, \ldots, n)$ is given by Eq (11).

Let $C_{\max}(k)$ be the makespan under a given k $(k = 0, 1, 2, \ldots, n)$, from the above analysis, we propose the following algorithm to solve the problem $1|ma, tpdde|C_{\max}$.

Algorithm 1

Step 1. Set $k = 0$, calculate $\mu_j = \Delta_j$ (see Eqs (6) and (7)) and $\nu_j = \theta_j a_j$, determine a local optimal job sequence by Lemma 1, and record the objective function $C_{\max}(k = 0)$.

Step 2. Set $k = 1, 2, \ldots, n-1$, calculate Δ_r (see Eq (11)) and determine a local optimal job sequence by an assignment problem Eqs (17)–(20), and record the objective function $C_{\max}(k)$ $(k = 1, 2, \ldots, n-1)$.

Step 3. Set $k = n$, calculate $\mu_j = \Delta_j$ (see Eqs (14) and (15)) and $\nu_j = a_j$, determine a local optimal job sequence by Lemma 1, and record the objective function $C_{\max}(k = n)$.

Step 4. The optimal job sequence is the one with the minimum objective function value $C^*_{\max} = \min\{C_{\max}(k)|k = 0, 1, 2, \ldots, n\}$.

Theorem 1. *The problem* $1|ma, tpdde|C_{\max}$ *can be solved by Algorithm 1 in* $O(n^4)$ *time.*

Proof. Steps 1 and 3 need $O(n \log n)$ time respectively. For each k ($k = 1, 2, \ldots, n-1$), the running time for solving each assignment problem needs $O(n^3)$ time. Step 4 needs $O(n)$ time. Hence, the overall time complexity of Algorithm 1 is $O(n^4)$ time. □

Example 1. *Consider a* 6*-job problem* $1|ma, tpdde|C_{\max}$*, where* $a_1 = 3, a_2 = 4, a_3 = 5, a_4 = 8, a_5 = 9, a_6 = 7$*,* $\theta_1 = 0.7$*,* $\theta_2 = 0.6$*,* $\theta_3 = 0.7$*,* $\theta_4 = 0.8$*,* $\theta_5 = 0.6$*,* $\theta_6 = 0.9$*,* $\tilde{t} = 1$*,* $b = 0.15$*,* $c = 0.3$*, and* $D = 3$*.*

Solution:

For $k = 0$, from Eqs (7) and (8), we have $\Delta_1 = 2.7453$, $\Delta_2 = 2.8530$, $\Delta_3 = 2.6660$, $\Delta_4 = 2.3680$, $\Delta_5 = 2.0368$, $\Delta_6 = 1.7118$, and $E = 5.1571$. The local optimal job sequence is $\varphi = \{J_2 \to J_1 \to J_3 \to J_5 \to J_6 \to J_4\}$, and $C_{\max}(0) = 63.6427$.

For $k = 1$, from Eqs (11) and (12), we have $\Delta_1 = 1$, $\Delta_2 = 2.1845$, $\Delta_3 = 2.2701$, $\Delta_4 = 2.1214$, $\Delta_5 = 1.8842$, $\Delta_6 = 1.6207$, and $E = 4.6621$. The local optimal job sequence is $\varphi = \{J_6 \to J_2 \to J_1 \to J_3 \to J_5 \to J_4\}$, and $C_{\max}(1) = 49.6442$.

For $k = 2$, from Eqs (11) and (12), we have $\Delta_1 = 1.1847$, $\Delta_2 = 1.2311$, $\Delta_3 = 1.7573$, $\Delta_4 = 1.8262$, $\Delta_5 = 1.7065$, $\Delta_6 = 1.5157$, and $E = 4.3832$. The local optimal job sequence is $\varphi = \{J_4 \to J_6 \to J_2 \to J_1 \to J_3 \to J_5\}$, and $C_{\max}(2) = 44.6886$.

For $k = 3$, from Eqs (11) and (12), we have $\Delta_1 = 1.4317$, $\Delta_2 = 1.4879$, $\Delta_3 = 1.3904$, $\Delta_4 = 1.4317$, $\Delta_5 = 1.4879$, $\Delta_6 = 1.3904$, and $E = 4.2930$. The local optimal job sequence is $\varphi = \{J_3 \to J_1 \to J_6 \to J_5 \to J_2 \to J_4\}$, and $C_{\max}(3) = 45.8487$.

For $k = 4$, from Eqs (11) and (12), we have $\Delta_1 = 1.7573$, $\Delta_2 = 1.8262$, $\Delta_3 = 1.7065$, $\Delta_4 = 1.5157$, $\Delta_5 = 1.1847$, $\Delta_6 = 1.2311$, and $E = 4.3832$. The local optimal job sequence is $\varphi = \{J_2 \to J_1 \to J_3 \to J_6 \to J_4 \to J_5\}$, and $C_{\max}(4) = 50.2634$.

For $k = 5$, from Eqs (11) and (12), we have $\Delta_1 = 2.1845$, $\Delta_2 = 2.2701$, $\Delta_3 = 2.1214$, $\Delta_4 = 1.8842$, $\Delta_5 = 1.6207$, $\Delta_6 = 1$, and $E = 4.6621$. The local optimal job sequence is $\varphi = \{J_2 \to J_1 \to J_3 \to J_6 \to J_4 \to J_5\}$, and $C_{\max}(5) = 62.3724$.

For $k = 6$, from Eqs (15) and (16), we have $\Delta_1 = 2.7453$, $\Delta_2 = 2.8530$, $\Delta_3 = 2.6660$, $\Delta_4 = 2.3680$, $\Delta_5 = 2.0368$, $\Delta_6 = 1.7118$, and $E = 5.1571$. The local optimal job sequence is $\varphi = \{J_2 \to J_1 \to J_3 \to J_6 \to J_4 \to J_5\}$, and $C_{\max}(6) = 86.3039$.

From above analysis, the optimal value is $k = 2$, the optimal job sequence is $\varphi^* = \{J_4 \to J_6 \to J_2 \to J_1 \to J_3 \to J_5\}$, and the optimal value is $C^*_{\max} = 44.6886$.

4. Conclusions

We studied the single-machine problem with *tpdde* and *ma*, where the objective function is to minimize the makespan. It is showed that the problem $1|ma, tpdde|C_{\max}$ can be solved in $O(n^4)$ time. Future research may focus on the problems with general time-and-position dependent deteriorating effects $p_{[j]} = \begin{cases} (a_{[j]} + b_{[j]} s_{[j]}) j^c, & \text{if } j \le k, \\ (\theta_{[j]} a_{[j]} + b_{[j]}(s_{[j]} - C_{[k]} - D))(j - k)^c, & \text{if } j > k. \end{cases}$ Another possible challenging is to consider the problems under parallel machines. The model assumptions can also be extended to for several special cases of processing set restrictions like as Scenario-Dependent Component Processing Times or Release Dates (see Wu et al. [31]; Wu et al. [32]; Wu et al. [33]).

Acknowledgments

This research was supported by the National Natural Science Regional Foundation of China (71861031 and 72061029).

Conflict of interest

The authors declare that they have no conflicts of interest.

References

1. J. B. Wang, M. Z. Wang, Minimizing makespan in three-machine flow shops with deteriorating jobs, *Comput. Oper. Res.*, **40** (2013), 547–557. http://dx.doi.org/10.1016/j.cor.2012.08.006
2. J. B. Wang, J. J. Wang, Single machine group scheduling with time dependent processing times and ready times, *Inform. Sci.*, **275** (2014), 226–231. http://dx.doi.org/10.1016/j.ins.2014.02.034
3. J. B. Wang, J. J. Wang, Single-machine scheduling problems with precedence constraints and simple linear deterioration, *Appl. Math. Modell.*, **39** (2015), 1172–1182. http://dx.doi.org/10.1016/j.apm.2014.07.028
4. T. C. E. Cheng, S. C. Tseng, P. J. Lai, W. C. Lee, Single-machine scheduling with accelerating deterioration effects, *Optim. Lett.*, **8** (2014), 543–554. https://doi.org/10.1007/s11590-012-0539-8
5. Y. Yin, T. C. E. Cheng, L. Wan, C. C. Wu, J. Liu, Two-agent single-machine scheduling with deteriorating jobs, *Comput. Ind. Eng.*, **81** (2015), 177–185. https://doi.org/10.1016/j.cie.2015.01.002
6. X. Zhang, W. C. Lin, W. H. Wu, C. C. Wu, Single-machine common/slack due window assignment problems with linear decreasing processing times, *Eng. Optim.*, **49** (2017), 1388–1400. https://doi.org/10.1080/0305215X.2016.1248180
7. F. Liu, J. Yang, Y. Y. Lu, Solution algorithms for single-machine group scheduling with ready times and deteriorating jobs, *Eng. Optim.*, **51** (2019), 862–874. https://doi.org/10.1080/0305215X.2018.1500562
8. J. B. Wang, X. X. Liang, Group scheduling with deteriorating jobs and allotted resource under limited resource availability constraint, *Eng. Optim.*, **51** (2019), 231–246. https://doi.org/10.1080/0305215X.2018.1454442
9. S. Gawiejnowicz, Models and algorithms of time-dependent scheduling, Springer, Berlin, 2020.
10. Y. Ma, C. Chu, C. Zuo, A survey of scheduling with deterministic machine availability constraints, *Comput. Ind. Eng.*, **58** (2010), 199–211. http://dx.doi.org/10.1016/j.cie.2009.04.014
11. J. B. Wang, C. M. Wei, Parallel machine scheduling with a deteriorating maintenance activity and total absolute differences penalties, *Appl. Math. Comput.*, **177** (2011), 8093–8099. http://dx.doi.org/10.1016/j.amc.2011.03.010

12. C. J. Hsu, T. C. E. Cheng, D. L. Yang, Unrelated parallel-machine scheduling with rate-modifying activities to minimize the total completion time, *Inform. Sci.*, **181** (2015), 4799–4803. http://dx.doi.org/10.1016/j.ins.2011.06.010

13. P. Ji, G. Li, Y. Huo, J. B. Wang, Single-machine common flow allowance scheduling with job-dependent aging effects and a deteriorating maintenance activity, *Optim. Lett.*, **8** (2014), 1389–1400. http://dx.doi.org/10.1007/s11590-012-0504-6

14. K. Rustogi, V. A. Strusevich, Combining time and position dependent effects on a single machine subject to rate modifying activities, *Omega*, **42** (2014), 166–178. http://dx.doi.org/10.1016/j.omega.2013.05.005

15. K. Rustogi, V. A. Strusevich, Single machine scheduling with time-dependent linear deterioration and rate-modifying maintenance, *J. Oper. Res. Soc.*, **66** (2015), 500–515. http://dx.doi.org/10.1057/jors.2014.18

16. C. L. Liu, Y. Fan, C. L. Zhao, J. J. Wang, Multiple common due-dates assignment and optimal maintenance activity scheduling with linear deteriorating jobs, *J. Ind. Manag. Optim.*, **13** (2017), 713–720. http://dx.doi.org/10.3934/jimo.2016042

17. X. Xiong, D. Wang, T. C. E. Cheng, C. C. Wu, Y. Yin, Single-machine scheduling and common due date assignment with potential machine disruption, *Int. J. Prod. Res.*, **56** (2018), 1345–1360. http://dx.doi.org/10.1080/00207543.2017.1346317

18. Z. Zhu, M. Liu, C. Chu, J. Li, Multitasking scheduling with multiple rate-modifying activities, *Intl. Trans. Oper. Res.*, **26** (2019), 1956–1976. https://doi.org/10.1111/itor.12393

19. X. Zhang, W. H. Wu, W. C. Lin, C. C. Wu, Machine scheduling problems under deteriorating effects and deteriorating rate-modifying activities, *J. Oper. Res. Soc.*, **69** (2018), 439–448. https://doi.org/10.1057/s41274-017-0200-0

20. J. B. Wang, L. Li, Machine scheduling with deteriorating jobs and modifying maintenance activities, *Comput. J.*, **61** (2018), 47–53. https://doi.org/10.1093/comjnl/bxx032

21. X. Zhang, S. C. Liu, W. C. Lin, C. C. Wu, Parallel-machine scheduling with linear deteriorating jobs and preventive maintenance activities under a potential machine disruption, *Comput. Ind. Eng.*, **145** (2020), 106482. https://doi.org/10.1016/j.cie.2020.106482

22. X. Zhang, D. Y. Bai, W. C. Lin, S. R. Cheng, C. C. Wu, Single-machine slack due-window assignment scheduling with multi-maintenance activities and position-and-resource-dependent processing times, *Int. J. Syst. Sci. Oper. Logist.*, **2021** (2021), 1–12. https://doi.org/10.1080/23302674.2021.1921305

23. X. Y. Sun, X. N. Geng, Single machine scheduling with deteriorating effects and machine maintenance, *Int. J. Prod. Res.*, **57** (2019), 3186–3199. https://doi.org/10.1080/00207543.2019.1566675

24. J. B. Wang, Y. Hu, B. Zhang, Common due-window assignment for single-machine scheduling with generalized earliness/tardiness penalties and a rate-modifying activity, *Eng. Optim.*, **53** (2021), 496–512. https://doi.org/10.1080/0305215X.2020.1740921

25. X. Jia, D. Y. Lv, Y. Hu, J. B. Wang, Z. Wang, E. Wang, Slack due-window assignment scheduling problem with deterioration effects and a deteriorating maintenance activity, *Asia Pac. J. Oper. Res.*, **2022** (2022), forthcoming. https://doi.org/10.1142/S0217595922500051
26. H. He, Y. Hu, W. W. Liu, Scheduling with deteriorating effect and maintenance activities under parallel processors, *Eng. Optim.*, **53** (2021), 2070–2087. https://doi.org/10.1080/0305215X.2020.1844194
27. S. Eilon, On a mechanistic approach to fatigue and rest periods, *Int. J. Prod. Res.*, **3** (1964), 327–332. https://doi.org/10.1080/00207546408943065
28. E. J. Lodree, C. D. Geiger, A note on the optimal sequence position for a rate-modifying activity under simple linear deterioration, *Eur. J. Oper. Res.*, **201** (2010), 644–648. https://doi.org/10.1016/j.ejor.2009.03.027
29. M. Ji, C. J. Hsu, D. L. Yang, Single-machine scheduling with deteriorating jobs and aging effects under an optional maintenance activity consideration, *J. Comb. Optim.*, **26** (2013), 437–447. https://doi.org/10.1007/s10878-011-9415-1
30. G. H. Hardy, J. E. Littlewood, G. Polya, *Inequalities*, Cambridge University Press, 1988.
31. C. C. Wu, J. N. D. Gupta, S. R. Cheng, B. M. T. Lin, S. H. Yip, W. C. Lin, Robust scheduling of a two-stage assembly shop with scenario-dependent processing times, *Int. J. Prod. Res.*, **59** (2021), 5372–5387. https://doi.org/10.1080/00207543.2020.1778208
32. C. C. Wu, A. Azzouz, J. Y. Chen, J. Xu, W. L. Shen, L. Lu, et al., A two-agent one-machine multitasking scheduling problem solving by exact and metaheuristics, *Complex. Intell. Syst.*, **8** (2022), 199–212. https://doi.org/10.1007/S40747-021-00355-4
33. C. C. Wu, D. Bai, X. Zhang, S. R. Cheng, W. C. Lin, Z. L. Wu, A robust customer order scheduling problem along with scenario-dependent component processing times and due dates, *J. Manuf. Syst.*, **58** (2021), 291–305. https://doi.org/10.1016/j.jmsy.2020.12.013