Wenhui Yang, Lu Chen & Stèphane Dauzère-Pèrés (2022) A dynamic optimisation approach for a single machine scheduling problem with machine conditions and maintenance decisions, International Journal of Production Research, 60:10, 3047-3062, DOI: 10.1080/00207543.2021.1910746

A dynamic optimisation approach for a single machine scheduling problem with machine conditions and maintenance decisions

Wenhui Yang[a] Lu Chen[a,*] Stéphane Dauzère-Pérès[b,c]

[a] School of Mechanical Engineering,
Shanghai Jiao Tong University, Shanghai, China, 200240
[b] Department of Manufacturing Sciences and Logistics,Mines Saint-Etienne,
Univ. Clermont Auvergne, CNRS, UMR 6158 LIMOS, CMP, Gardanne, France
[c] Department of Accounting, Auditing and Business Analytics,
BI Norwegian Business School, 0484 Oslo, Norway

Abstract

In modern production systems, considering machine conditions is becoming essential to achieve an overall optimisation of the production schedule. This paper studies a single machine scheduling problem, where the actual processing times of jobs depend on their position in the production sequence and maintenance is considered. Moreover, the machine is subject to an uncertain condition variation. There is a tradeoff between rejecting a maintenance action, resulting in longer processing times, and accepting a maintenance action, leading to higher processing efficiency for future jobs. The problem is formulated as a finite-horizon Markov Decision Process. The objective is to minimize the makespan. Optimality properties are analyzed, based on which a dynamic optimisation approach is developed. Computational experiments demonstrate the effectiveness of the proposed approach.

Keywords : Single machine scheduling; Machine condition; Markov Decision Process; Dynamic optimisation approach; Maintenance

1 Introduction

Numerous studies on scheduling problems assume that machines process jobs in a *perfect* condition. However, this is never the case in real workshops. For instance, a machine may be shut down deliberately for a routine maintenance or unexpectedly by a random breakdown. Once one of these situations occurs, the loss of production can rather easily be quantified (Allahverdi and Mittenthal, 1994, 1995; Kao et al., 2018). In practice, there also exists invisible machine conditions between *perfect* and *down* due to loose linkage, wear and fatigue of parts, or misalignment of tighteners, etc (Lee et

*Corresponding author: Email: chenlu@sjtu.edu.cn

al., 2013; Ahmadzadeh and Lundberg, 2013; Huang et al., 2019). How to measure the impact of invisible machine conditions on scheduling is challenging.

The single machine scheduling problem studied in this paper is derived from a real valve workshop in *China Astronautics*. The valve production is a typical multi-variety small-batch production. Different types of valve share same resources (i.e. the milling process of valve hosing parts, the detecting process of valve shell shape) in the workshop. It is thus important to ensure a high throughput for those shared resources that are usually bottlenecks. Moreover, it is observed that the later a job is processed, the longer the processing time of the job to meet quality requirements. In this case, a maintenance action is usually conducted to increase the efficiency of the machine. The decision to be made here is a joint sequence of jobs in a batch and maintenance actions on critical resources. The objective is to minimize the makespan. There is a tradeoff between rejecting a maintenance action, resulting in longer processing times for future jobs, and accepting a maintenance action, to ensure a higher processing efficiency of the machine. With the development of sensor technology and Prognostics and Health Management, some statistics during production can be collected, based on which the reliability or the lifetime of machines can be derived. However, this is still not possible or too complex in many manufacturing contexts, and a deterministic machine deterioration process is assumed in most studies. We consider in our paper an uncertain machine condition variation. In the workshop, the machine condition is categorized into three levels, namely stable, unstable, and down. A finite-horizon Markov Decision Process (MDP) is applied to describe the dynamic and uncertain evolution of the invisible machine condition. An optimal joint sequence of jobs and maintenance actions can be obtained by studying optimality properties of the problem.

The paper is organized as follows. Section 2 reviews the related literature. In Section 3, a single machine scheduling model is proposed based on a MDP. A dynamic optimisation approach is developed in Section 4. Section 5 presents and analyzes computational results. Finally, Section 6 provides concluding remarks and some perspectives on future research.

2 Literature review

We first review works on scheduling with machine condition constraints, followed by the literature on joint optimisation of scheduling and maintenance. Our scientific contributions are then discussed.

2.1 Scheduling with machine condition constraints

In production scheduling, reliability and availability are the two major measurements to quantify machine condition.

There are three typical ways to define reliability, namely experienced-based approach, model-based approach, and data-driven approach (Gorjian et al., 2010). An experiencebased approach usually defines machine reliability based on the mathematical distribution, such as Poisson distribution, Weibull distribution (Yan and Hua, 2010), and Exponential distribution (Johnson, 1989). Najid et al. (2011) and Yildirim and Nezami (2014) analyzed production plainning problems constrained by machine reliability based on non-homogenous Poisson distribution. Castanier et al. (2015) studied a production scheduling problem, which is influenced by stochastic failures modeled by Weibull distribution. Zhou et al. (2017) addressed a reliability constrained task scheduling problem to minimize makespan where the machine failures followed an exponential law.

When historical repair and failure data are missing, an experienced-based approach cannot be used. A model-based approach monitors machine reliability by a physics-based model or a statistical model. For instance, Han et al. (2016) and Shui et al. (2019) developed models that incorporate both physical models and data analytics to represent the interactions among operational reliability and product characteristics. Regression models (Li et al., 2017) and functional quantitative / qualitative models (Sun et al., 2017) are some typically statistical models to measure machine reliability during production process.

A data-driven approach quantifies machine reliability based on recognizing pattern variation (Regattieri et al., 2010). Sensor data is used to estimate and predict reliability (Alsina et al., 2017; Tian, et al., 2018). Abundant historical data is needed to support a specific variation rule of machine reliability.

Machine availability is another widely used measurement to define machine condition. Lee (1996) firstly addressed scheduling with a single fixed unavailable period. It was assumed that the start time and the duration of the unavailable period were known in advance. These assumptions are too idealistic in real application. Ji et al. (2007) found that routine maintenance brings periodic unavailable periods. Since then, machine scheduling with periodic machine availability were studied (Low et al., 2010; Yu et al., 2014; Gonzalez and Framinan, 2018; Nesello et al., 2018). Tamssaouet et al. (2018) used an approach based on disjunctive graphs to represent job sequences when solving the job-shop scheduling problem with fixed unavailability periods. Dieulle et al. (2003) found that the machine becomes unavailable when its condition drops below a pre-set critical threshold. This finding leads to a more realistic situation with flexible unavailability intervals. Nourelfath et al. (2010) and Peng and van Houtum (2016) studied a minimum production cost scheduling problem with flexible unavailable production periods. Lee and Kim (2015) studied a parallel machine scheduling problem with production availability constraints, where production availability refers to the physical limitation of machines. For example, some jobs cannot be processed on a specific machine.

Some other studies used the processing efficiency or the product yield rate to define the machine condition. Martinelli (2005) proposed optimal production policies constrained by production-dependent failures. Sloan (2004) defined a product yield rate that depends on the machine condition following a binomial distribution. Cheng et al. (2016) studied a joint optimisation of the production rate and maintenance in machining systems where the production-dependent failure rate is modeled by a Weibull cumulative damage model. Broek et al. (2020) adjusted the production rates to balance the output and the failure risk in condition-based production planning. They assumed that deterioration increments are fixed for a given production rate.

In the literature, it has already been discussed that the efficiency of schedules is heavily influenced by the machine condition. However, a deterministic machine deterioration process is assumed in most studies. To define the uncertain machine condition variation, Markov chains are usually considered to model the machine deterioration from the state "as good as new" to the state "breakdown" (Neves et al., 2011). In this case, the machine condition is classified by a finite number of discrete states whose variation is governed by the state transition (Boukas and Liu, 2001). The efficiency and effectiveness of the method was demonstrated by successfully applying it to real data (Kim et al, 2011). The numerical results in Khaleghei and Makis (2015) validated the robustness of the Markov model. More precisely, even when the historical repair and failure data are missing, the model can still be effective. Kurt and Kharoufeh (2009) applied a Markovian model in a scheduling problem and proposed an optimal maintenance policy. Liu et al. (2018) used a hidden semi-Markov model to predict the machine condition and to integrate maintenance actions in a single machine scheduling problem.

2.2 Joint optimisation of scheduling and maintenance

Maintenance results in the update of the machine condition. Lee (1997) first introduced maintenance into production scheduling. Since then, the integration of scheduling and maintenance has been extensively studied (Boudjelida, 2019). Cassady and Kutanoglu (2005) proposed an integrated model for a single machine scheduling problem with preventive maintenance to minimize the weighted sum of the completion times. Mokhtari et al. (2012) introduced multiple maintenance services in a parallel machine scheduling system and developed a population-based variable neighborhood search algorithm. Yang (2013) investigated scheduling problems with deterioration effects, in which they assumed that the duration of maintenance activities depends on the machine running time and the processing time of jobs varies according to the machine condition. Yildirim and Nezami (2014) proposed a single machine scheduling model with preventive maintenance, which is triggered when the reliability is lower than a threshold. Ting et al. (2018) studied a single machine scheduling problem with flexible periodic maintenance. Maintenance activities keep the machine condition acceptable by reverting the machine from a sub-normal processing rate to a normal one.

2.3 Summary and contributions

Our work on the single machine scheduling problem with discrete machine condition constraints contributes to a new field of research. The consideration of machine conditions has been shown in numerous studies to help improve the operational performance in production scheduling. There are also a large number of studies on the joint optimisation of production scheduling and preventive maintenance. However, considering a machine condition variation that is uncertain and its impact on job processing times has been explored on a very limited scale. The contributions of our paper are summarized below:

1. A finite-horizon MDP model is developed to simultaneously consider the machine condition transition and maintenance activities. The joint optimisation of the production sequence of jobs and of maintenance decisions can be achieved.
2. Two optimality properties are proposed, based on which a dynamic optimisation algorithm is developed to solve real size instances. Moreover, a rule-based initial solution generation procedure is embedded to reduce the complexity of the dynamic algorithm.
3. Real data collected from a valve workshop are used to validate the performance of the proposed dynamic algorithm. Useful managerial insights regarding the impact of the uncertain machine condition variation on scheduling are derived.

3 Mathematical formulation

3.1 Problem description

The bathtub curve has been widely used to describe the lifetime of a population of products (Xie et al., 2002). The bathtub curve consists of three periods: An infant mortality period with a decreasing failure rate; a normal life period with a constant failure rate; and a wear-out period with an increasing failure rate.

The machine addressed in this paper is at the middle stage of the bathtub curve, and its reliability is at a stable level with a constant failure rate. As processing continues, the processing time of jobs increase due to reasons as machine tools temperature change and abrasion etc. Let t_i be the nominal value of the processing time for job i, and t_{ij} be the actual processing time when job i is scheduled in the j^{th} position. We define t_{ij} according to the sum-of-processing-times-based deteriorating model (Cheng et al., 2010) as:

$$t_{ij} = t_i\left(1 - \frac{\sum_{l=1}^{j-1} t_{[l]}}{\sum_{l=1}^{n} t_l}\right)^{-\frac{1}{2}} = t_i\sqrt{\frac{\sum_{l=1}^{n} t_l}{\sum_{l=j}^{n} t_{[l]}}} \tag{1}$$

where $t_{[l]}$ is the nominal processing time of the job processed in the l^{th} position in the sequence of jobs.

In addition, we consider that the machine has two working states correlated through the transition probabilities. Machine breakdowns are not considered in this study. Let us denote the two states 0 (stable) and 1 (unstable). Their transition probabilities satisfy $p(0|0) = 1 - p(1|0)$ and $p(1|1) = 1 - p(0|1)$, as shown in Figure 1. When a maintenance action occurs, the transition probabilities are updated to be $p_m(0|0)$ and $p_m(1|1)$.

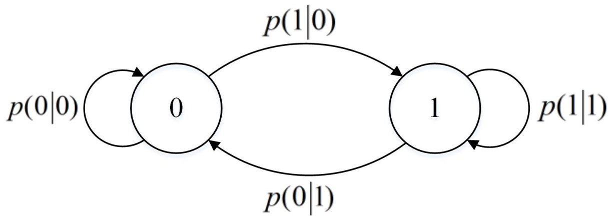


Figure 1: Transition probabilities between machine states

When the machine is in the unstable state, the processing time is observed to be longer. The unstable state can be associated with a deteriorating rate d, $0 < d < 1$. Let $t_{ij}^{'}$ be the processing time of job i in the j^{th} position when the machine is in the unstable state.

$$t_{ij}^{'} = t_{ij}(1+d) \tag{2}$$

With (1) and (2):

$$t_{ij}^{'} = t_{ij}(1+d) = t_i(1+d)\sqrt{\frac{\sum_{l=1}^{n} t_l}{\sum_{l=j}^{n} t_{[l]}}} \tag{3}$$

The scheduling problem can be described as follows. A set of N jobs needs to be processed on a single machine. After the machine completes a job, the machine state has a probability to stay in its current state or to move to the other state. The processing time of a given job depends on its position in the sequence and on the machine state. A maintenance action (MA) can be inserted between any two jobs in the sequence. In this study, we consider only minor non-replacement maintenance actions (Makis et al., 1998; Liu et al., 2018) on the machine, in which lubrication, adjustment or cleaning may happen. After an MA, the probability of the machine remaining stable is higher. That is, $p(0|0) < p_m(0|0)$ and $p(1|1) > p_m(1|1)$. Thus, the expected processing time of the job after the MA is shorter.

The problem is to find a job sequence and the corresponding maintenance decisions, such that: (1) Each job is assigned to only one position; (2) Positions are consecutive and each position is assigned to only one job; (3) The machine processes one job at a time. The objective is to minimize the makespan.

The decision variables are

x_{ij}^{0} $= 1$, if job i is processed in the j^{th} position, and no MA is inserted at the beginning of position j; $= 0$, otherwise. $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, N$.

x_{ij}^{1} $= 1$, if job i is processed in the j^{th} position, and an MA is inserted at the beginning of position j; $= 0$, otherwise. $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, N$.

Let I be the job sequence $I = (i_1, i_2, \ldots, i_N)$, and Y be the MA sequence $Y = (y_1, y_2, \ldots, y_N)$. For position j, if $x_{ij}^{1} = 1$ $(i = 1, 2, \ldots, N)$, $y_j = 1$; otherwise, $y_j = 0$. The optimal solution is a joint sequence of I and Y (denoted as $\{I^*, Y^*\}$) that minimizes the makespan MaxC_j, where C_j is the completion time of the job at position j (j = 1, 2, ..., N).

3.2 Model development

Before defining the mathematical model, we first present two Lemmas to calculate C_j. It is assumed that the machine starts from the stable state. That is, $C_1 = t_i x_{i1}^0$.

An arbitrary position n in the job sequence has two times: The processing time of the job at position n and the duration of the MA (if there is an MA between the job at position n-1 and the job at position n). And the following four situations may take place.

(1) The machine is in the stable state at the $(n-1)^{th}$ position and there is no MA between position $n-1$ and position n. C_n can be calculated as:

$$\begin{aligned} C_n &= C_{n-1} + p(0|0)t_{in}x_{in}^0 + (1-p(0|0))t_{in}^{'}x_{in}^0 \\ &= C_{n-1} + (1-p(0|0)d + d)t_{in}x_{in}^0 \end{aligned} \tag{4}$$

(2) The machine is in the stable state at the $(n-1)^{th}$ position and there is an MA between position $n-1$ and position n. C_n can be calculated as:

$$\begin{aligned} C_n &= C_{n-1} + b + p_m(0|0)t_{in}x_{in}^1 + (1-p_m(0|0))t_{in}^{'}x_{in}^1 \\ &= C_{n-1} + b + (1-p_m(0|0)d + d)t_{in}x_{in}^1 \end{aligned} \tag{5}$$

(3) The machine is in the unstable state at the $(n-1)^{th}$ position and there is no MA between position $n-1$ and position n. C_n can be calculated as:

$$\begin{aligned} C_n &= C_{n-1} + (1-p(1|1))t_{in}x_{in}^0 + p(1|1)t_{in}^{'}x_{in}^0 \\ &= C_{n-1} + (1+p(1|1)d)t_{in}x_{in}^0 \end{aligned} \tag{6}$$

(4) The machine is in the unstable state at the $(n-1)^{th}$ position and there is an MA between position $n-1$ and position n. C_n can be calculated as:

$$\begin{aligned} C_n &= C_{n-1} + b + (1-p_m(1|1))t_{in}x_{in}^1 + p_m(1|1)t_{in}^{'}x_{in}^1 \\ &= C_{n-1} + b + (1+p_m(1|1)d)t_{in}x_{in}^1 \end{aligned} \tag{7}$$

Equations (4) to (7) can be written in the following general form:

$$C_n = C_{n-1} + by_n + \Delta t_{in}x_{in} = \begin{cases} C_{n-1} + by_n + \delta_1 t_{in}x_{in}^0, & \delta_1 = 1 - p(0|0)d + d \\ C_{n-1} + by_n + \delta_2 t_{in}x_{in}^1, & \delta_2 = 1 - p_m(0|0)d + d \\ C_{n-1} + by_n + \delta_3 t_{in}x_{in}^0, & \delta_3 = 1 + p(1|1)d \\ C_{n-1} + by_n + \delta_4 t_{in}x_{in}^1, & \delta_4 = 1 + p_m(1|1)d \end{cases} \tag{8}$$

where Δ is a coefficient that depends on the situation. It can be observed that, if an MA is inserted, it can reduce the prolonged time of the job processing time caused by the unstable state of the completion time of the position.

In order to reduce the exponential number of possibilities that need to be considered, the following two lemmas are proposed to identify optimal positions to insert an MA.

Lemma 1:An MA needs to be inserted between position $j-1$ and position j ($j = 2, 3, \ldots, N$) in an optimal solution, if: 1) the machine is in the stable state at the end of the $(j-1)^{th}$ position, and 2) $b < (p_m(0|0) - p(0|0))dt_{ij}$.

Proof:

Let $C_j^{'}$ be the completion time at the j^{th} position with an MA between position $j-1$ and position j, and $C_j^{''}$ be the completion time at the j^{th} position without an MA. From equation (4) and (5), the difference between $C_j^{'}$ and $C_j^{''}$ is,

$$\begin{aligned}\Delta C &= C_j^{'} - C_j^{''} \\ &= b + p_m(0|0)t_{ij} + (1 - p_m(0|0))t_{ij}^{'} - p(0|0)t_{ij} - (1 - p(0|0))t_{ij}^{'} \\ &= b - (p_m(0|0) - p(0|0))dt_{ij}.\end{aligned}$$

With $p_m(0|0) - p(0|0) > 0$, $b < (p_m(0|0) - p(0|0))dt_{ij}$, $\Delta C < 0$.

Thus, inserting an MA between position $j-1$ and position j can reduce the completion time at position j.

Lemma 1 has been proven.

Lemma 2:An MA needs to be inserted between position $j-1$ and position j ($j = 3, 4, \ldots, N$) in an optimal solution, if: 1) the machine is in the unstable state at the end of the $(j-1)^{th}$ position, and 2) $b < (p(1|1) - p_m(1|1))dt_{ij}^{'}$.

Lemma 2 can be proved in a similar way to that of **Lemma 1**.

The above two MA decision rules give sufficient conditions to insert MAs in optimal solutions. Thus, the number of feasible solutions calculated by a typical backward recursive method is reduced from $4^{N-1}A_N^{N-1}$ to $2^{N-2}A_N^{N-1}$ ($N \geq 2$).

Denote Z as the objective function value. Based on the above analysis, a finite-horizon MDP model can be formulated as:

$$\text{Minimize} \quad Z = \text{Max} C_j \tag{9}$$

subject to:

$$\sum_{i=1}^{N} x_{ij}^0 + \sum_{i=1}^{N} x_{ij}^1 = 1, j = 1, 2, ..., N \tag{10}$$

$$\sum_{j=1}^{N} x_{ij}^0 + \sum_{i=1}^{N} x_{ij}^1 = 1, i = 1, 2, ..., N \tag{11}$$

$$\begin{aligned}C_j =& C_{j-1} + x_{ij}^0 min\left\{[(p(0|0)t_{ij} + (1 - p(0|0))t_{ij}^{'}], [(1 - p(1|1))t_{ij} + p(1|1)t_{ij}^{'}]\right\} \\ &+ x_{ij}^1 min\left\{[b + p_m(0|0)t_{ij} + (1 - p_m(0|0))t_{ij}^{'}], [b + (1 - p_m(1|1))t_{ij} + p_m(1|1)t_{ij}^{'}]\right\}, \\ & j = 2, 3, ..., N\end{aligned} \tag{12}$$

$$x_{ij}^{0} \in \{0,1\}, x_{ij}^{1} \in \{0,1\}, i = 1,2,...,N, j = 1,2,...,N \tag{13}$$

The objective function (9) minimizes the makespan. Constraints (10) ensure that each job i is assigned to exactly one position. Constraints (11) ensure that each position j is assigned to only one job. Constraints (12) calculate the minimum processing time of job i which is processed in position j. Constraints (13) define the domains of the decision variables.

3.3 Optimal objective function value calculation

A Backward Recursive Method (BAM) is used to calculate the makespan of the above finite-horizon MDP model. The following notations are used in the BAM:

$\{I_j, Y_j\}$	Partial solution from position j $(j = 1,2,...,N)$.
$t_{i,j-1}(I_j, Y_j)$	Expected processing time at position $j-1$ when the job i $(i = 1,2,...,N)$ is assigned to position $j-1$ with $\{I_j, Y_j\}$ as the partial solution from position j $(j = 2,3,...,N)$. $C_{i,N+1} = 0$.
$C_j(I_j, Y_j)$	Expected completion time at position N with $\{I_j, Y_j\}$ as the partial solution from position j $(j = 1,2,...,N)$.

When $j = 1$, $Z = C(I_1, Y_1)$ is the makespan of all the jobs of solution $\{I, Y\}$.

The BAM calculates the completion time at each position from the last position and is described in **Algorithm 1** (shown in Table 1).

Table 1: The algorithm of the backward recursive method

Algorithm 1. Backward recursive method
START
1. Initialize algorithm parameters $p(0\|0), p_m(0\|0), p(1\|1), p_m(1\|1), b, N$
2. Set $j = N$.
3. **While** $j > 1$ **do**
4. Set $i = 1$.
5. **While** $i \leq N$ **do**
6. Assign job i to position j.
7. Apply **Lemma 1** and **Lemma 2** to obtain y_j.
8. Calculate $t_{i,j}(I_{j+1}, Y_{j+1})$,
9. $C_j(I_j, Y_j) = C_{i,j-1}(I_{j-1}, Y_{j-1}) + \Delta t_{i,j}(I_{j+1}, Y_{j+1}) + by_j$
10. $i = i + 1$.
11. **end while**
12. $j = j - 1$.
13. **end while**
14. $Z^* = \min C(I, Y)$.
15. Return Z^*.
END

In line 7, the proposed two Lemmas are applied to make MA decisions for each position. The expected processing time at position j is calculated in line 8. In line 9, the completion time of the partial solution is updated. The optimal makespan Z^* is returned when the BAM terminates. The algorithm has a complexity of $O(\log N^2)$.

3.4 A numerical example

In this section, we present an illustrative example with ten jobs (from J1 to J10) to be processed. The nominal processing time (unit: min) for each job is: $t_1 = 166$, $t_2 = 142$, $t_3 = 143$, $t_4 = 174$, $t_5 = 122$, $t_6 = 132$, $t_7 = 158$, $t_8 = 145$, $t_9 = 138$, $t_{10} = 170$. The maintenance duration time b is 12min. The deteriorating rate d is set to 0.2.

Two production scenarios are considered. Their corresponding transition probabilities are given in Table 2. The machine has a higher probability to stay in a stable condition in scenario 1 than it does in scenario 2.

Table 2: Transition probabilities of the two scenarios

Scenario	$p(0\|0)$	$p(1\|0)$	$p(1\|1)$	$p(0\|1)$	$p_m(0\|0)$	$p_m(1\|0)$	$p_m(1\|1)$	$p_m(0\|1)$
1	0.78	0.22	0.60	0.40	0.85	0.15	0.10	0.90
2	0.55	0.45	0.70	0.25	0.72	0.28	0.35	0.65

Table 3 shows the results when the BAM is applied to obtain the optimal solutions for this instance.

Table 3: Optimal solutions for the instance

Scenario	$Z(min)$	$\{I^*, Y^*\}$
1	2430.6	$\{$(J8, J5, J6, J9, J2, J3, J7, J1, J10, J4), (0000000001)$\}$
2	2531.3	$\{$(J7, J5, J6, J9, J2, J3, J7, J1, J10, J4), (0000000011)$\}$

In Table 3, it can be observed that the optimal makespan under senario 2 is longer. And although the optimal sequences under the two scenarios are different, the partial sequences from the 2^{nd} position in both solutions are in non-decreasing order of the nominal processing times of the jobs.

Based on the above observation, we develop a dynamic optimisation approach, based on optimality properties, in which the enumeration procedure is replaced. Because it is very fast, the dynamic optimisation approach can be applied to solve real size instances.

4 A dynamic optimisation approach

In this section, two optimality properties are proposed firstly. We consider the situation where $p(0|0) + p(1|1) > 1$ and $p_m(0|0) + p_m(1|1) > 1$. It can be proved that our dynamic optimisation approach is also applicable to other situations.

4.1 Optimality properties

Remember that $C_j^{'}$ is the expected completion time at position j ($j = 1, 2, \ldots, N$) if the machine is stable at position j; and $C_j^{''}$ is the expected completion time at position j ($j = 1, 2, ..., N$) if the machine is unstable in position j. The following Lemma holds.

Lemma 3. With $p(0|0) + p(1|1) > 1$ and $p_m(0|0) + p_m(1|1) > 1$, $C_j^{'} < C_j^{''}$.

The proof is provided in the Appendix.

Lemma 3 can be extended to the situation where $p(0|0) + p(1|1) \leq 1$ and $p_m(0|0) + p_m(1|1) \leq 1$ as follows.

Lemma 4. With $p(0|0) + p(1|1) \leq 1$ and $p_m(0|0) + p_m(1|1) \leq 1$, $C_j^{'} < C_j^{''}$.

Based on **Lemma 3** and **Lemma 4**, the following proposition can be derived.

Proposition 1. In an optimal solution, the partial sequence from the 2^{nd} position follows the Shortest Processing Time (SPT) rule.

Proof

Assume that jobs h, i and k are three adjacent jobs in an optimal sequence with job h sequenced in the $(r-1)^{th}$ position, job i sequenced in the r^{th} position and job k sequenced in the $(r+1)^{th}$ position ($r \geq 2$), and $t_h \leq t_i \leq t_k$. If exchanging jobs i and k does not decrease the expected processing times at the r^{th} and $(r+1)^{th}$ positions, the optimality is proved. Specifically, the following inequality needs to be proved.

$$(C_{i,r} - C_{h,r-1}) + (C_{k,(r+1)} - C_{i,r}) \leq (C_{k,r} - C_{h,r-1}) + (C_{i,(r+1)} - C_{k,r}) \tag{14}$$

The examination of inequality (14) starts by determining the optimal positions of the MAs based on **Lemma 1**. That can be done under the following three situations.

(1) $t_i \leq t_k \leq \frac{b}{(p_m(0|0) - p(0|0))d}$. Thus, there is no MA between position $r-1$ and position r, and no MA between position r and position $(r+1)$. (14) can be written as:

$$\begin{aligned} &(1 + d - p(0|0)d)t_i\sqrt{\frac{T}{T_r}} + (1 + d - p(0|0)d)t_k\sqrt{\frac{T}{T_r - t_i}} \\ &\leq (1 + d - p(0|0)d)t_k\sqrt{\frac{T}{T_r}} + (1 + d - p(0|0)d)t_i\sqrt{\frac{T}{T_r - t_k}} \end{aligned} \tag{15}$$

where $T = \sum_{j=1}^{n} t_i$ and $T_r = \sum_{r}^{n} t_{[r]}$. It is obvious that $1 + d - p(0|0)d > 1$, and (15) can be written as:

$$t_i\sqrt{\frac{T}{T_r}} + t_k\sqrt{\frac{T}{T_r - t_i}} \leq t_k\sqrt{\frac{T}{T_r}} + t_i\sqrt{\frac{T}{T_r - t_k}} \tag{16}$$

$$\frac{\sqrt{\frac{T}{T_r}} - \sqrt{\frac{T}{T_r - t_k}}}{\frac{t_k}{T}} \leq \frac{\sqrt{\frac{T}{T_r}} - \sqrt{\frac{T}{T_r - t_i}}}{\frac{t_i}{T}} \tag{17}$$

The left-hand side of (17) is the slope of the straight line l_1 crossing the points $(x_1, y_1) = ((\frac{T_r}{T} - \frac{t_k}{T}), \sqrt{\frac{T}{T_r} - \frac{T}{t_k}})$ and $(x_3, y_3) = (\frac{T_r}{T}, \sqrt{\frac{T}{T_r}})$. Similarly, the right-hand side of (17) is the slope of the straight line l_2 crossing the points $(x_2, y_2) = ((\frac{T_r}{T} - \frac{t_i}{T}), \sqrt{\frac{T}{T_r} - \frac{T}{t_i}})$ and (x_3, y_3). Since $f(x) = x^{-\frac{1}{2}}$ is monotone decreasing function of x for $x > 0$. It is obvious that $0 \leq x_1 \leq x_2 \leq x_3$. Thus, the slope of l_2 is greater than the slope of l_1 as shown in Figure 2. (17) is true.

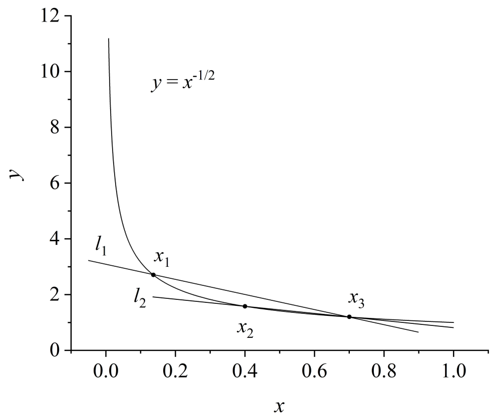


Figure 2: Illustration of the slope of lines

(2) $\frac{b}{(p_m(0|0)-p(0|0))d} \leq t_i \leq t_k$. Thus, an MA is inserted between position $r-1$ and position r, and an MA between position r and position $(r+1)$. (14) can be written as:

$$\begin{aligned} & b+(1+d-p_m(0|0)d)t_i\sqrt{\frac{T}{T_r}}+b+(1+d-p_m(0|0)d)t_k\sqrt{\frac{T}{T_r-t_i}} \\ & \leq b+(1+d-p_m(0|0)d)t_k\sqrt{\frac{T}{T_r}}+b+(1+d-p_m(0|0)d)t_i\sqrt{\frac{T}{T_r-t_k}} \end{aligned} \tag{18}$$

Since it is obvious that $1+d-p_m(0|0)d > 1$, (18) is true.

(3) $t_i \leq \frac{b}{(p_m(0|0)-p(0|0))d} < t_k$. Thus, there is no MA before job i is processed (in position r or position $r+1$) and an MA is inserted before job k is processed (in position r or position $r+1$). (14) can be written as:

$$\begin{aligned} & (1+d-p(0|0)d)t_i\sqrt{\frac{T}{T_r}}+b+(1+d-p_m(0|0)d)t_k\sqrt{\frac{T}{T_r-t_i}} \\ & \leq b+(1+d-p_m(0|0)d)t_k\sqrt{\frac{T}{T_r}}+(1+d-p(0|0)d)t_i\sqrt{\frac{T}{T_r-t_k}} \end{aligned} \tag{19}$$

Because it is obvious that $1+d-p(0|0)d > 0$, (19) can be written as:

$$\frac{\sqrt{\frac{T}{T_r}}-\sqrt{\frac{T}{T_r-t_k}}}{\frac{t_k}{T}} \leq \frac{1+d-p_m(0|0)d}{1+d-p(0|0)d}\frac{\sqrt{\frac{T}{T_r}}-\sqrt{\frac{T}{T_r-t_i}}}{\frac{t_i}{T}} \tag{20}$$

As $p(0|0) < p_m(0|0)$, $\frac{1+d-p_m(0|0)d}{1+d-p(0|0)d} < 1$, inequality (20) is true.

The above analysis proves that interchanging jobs i and k increases the expected processing times at the r^{th} and $(r+1)^{th}$ positions. The partial sequence from the 2^{nd} position of an optimal scheduling solution follows the SPT rule.

Proposition 1 has been proven.

In **Proposition 1**, it is assumed that $p(0|0)+p(1|1) > 1$ and $p_m(0|0)+p_m(1|1) > 1$. It can be proved that **Proposition 1** is also valid for any other scenarios of transition probabilities. For the sake of simplification, the proofs are not provided here.

4.2 Objective evaluation

With **Proposition 1**, potential job sequences can be enumerated by assigning an arbitrary job in the first position and the sequence from the 2^{nd} position follows the SPT rule. An optimal solution is the one with minimum Z.

Denote s_j to be the job sequence in which the j^{th} job in the SPT sequence is assigned in the first position. Let S be the set of job sequences, $S = (s_1, s_2, ..., s_N)$. It is obvious that s_1 is the SPT sequence. **Proposition 2** is proposed to find the optimal sequence (denote as s^*) in S.

Denote $Z(\text{SPT})$ as the makespan of the SPT sequence, and $Z(s_j)$ as the makespan of sequence s_j.

Proposition 2. If $Z(s_j) > Z(\text{SPT})$, then $Z(s_{j+1}) > Z(\text{SPT})$, $j \geq 2$.

Proof

From equation (8):

$Z(s_j) = t_{j1} + \Delta t_{j_1 2} + \Delta t_{j_2 3} + ... + \Delta t_{j_{(j-1)} j} + \Delta t_{j_{(j+1)}(j+1)} + ... + \Delta t_{j_N N}$

$Z(SPT) = t_{j_1 1} + \Delta t_{j_2 2} + \Delta t_{j_3 3} + ... + \Delta t_{j_{(j-1)}(j-1)} + \Delta t_{jj} + \Delta t_{j_{(j+1)}(j+1)} + ... + \Delta t_{j_N N}$

Thus,

$$
\begin{aligned}
\Delta Z &= Z(s_j) - Z(SPT) \\
&= t_{j1} - \Delta t_{jj} + (\Delta t_{j_1 2} - t_{j_1 1}) + (\Delta t_{j_2 3} - \Delta t_{j_2 2}) + ... + (\Delta t_{j_{(j-1)} j} - \Delta t_{j_{(j-1)}(j-1)}) \\
&= t_j - \Delta t_j \sqrt{\frac{T}{T - \sum_1^{j-1} t_{j_n}}} + [\Delta t_{j_1}(\sqrt{\frac{T}{T - t_j}} - 1) + \Delta t_{j_2}(\sqrt{\frac{T}{T - t_j - t_{j_1}}} - \sqrt{\frac{T}{T - t_{j_1}}}) + \\
&\quad ... + \Delta t_{j_{j-1}}(\sqrt{\frac{T}{T - \sum_1^{j-2} t_{j_n}} - t_j} - \sqrt{\frac{T}{T - \sum_1^{j-2} t_{j_n}}}) + (\Delta - 1) t_{j_1}] \\
&= \frac{T}{\Delta} - T\sqrt{\frac{T}{T - \sum_1^{j-1} t_{j_n}}} + t_{j_1} \frac{[\sqrt{\frac{T}{T-t_j}} - \sqrt{\frac{T}{T}}]}{\frac{t_j}{T}} + t_{j_2} \frac{[\sqrt{\frac{T}{T-t_j-t_{j_1}}} - \sqrt{\frac{T}{T-t_{j_1}}}]}{\frac{t_j}{T}} + ... + \\
&\quad t_{j_{j-1}} \frac{[\sqrt{\frac{T}{T-t_j-\sum_1^{j-2} t_{j_n}}} - \sqrt{\frac{T}{T-\sum_1^{j-2} t_{j_n}}}]}{\frac{t_j}{T}} + \frac{(\Delta - 1) t_{j_1 1}}{\frac{t_j}{T}} \\
&= \frac{T}{\Delta} - \sqrt{\frac{T}{T - \sum_1^{j-1} t_{j_n}}} + f(\frac{t_j}{T}) + \frac{(\Delta - 1) t_{j_1 1}}{\frac{t_j}{T}}
\end{aligned}
$$

where $T = \sum_{j=1}^{n} t_j$, $f(\frac{t_j}{T}) = t_{j_1} \frac{[\sqrt{\frac{T}{T-t_j}} - \sqrt{\frac{T}{T}}]}{\frac{t_j}{T}} + t_{j_2}[\frac{\sqrt{\frac{T}{T-t_j-t_{j_1}}}}{\frac{t_j}{T}} - \frac{\sqrt{\frac{T}{T-t_{j_1}}}}{\frac{t_j}{T}}] + ... +$ $t_{j_{j-1}} \frac{[\sqrt{\frac{T}{T-t_j-\sum_1^{j-2} t_{j_n}}} - \sqrt{\frac{T}{T-\sum_1^{j-2} t_{j_n}}}]}{\frac{t_j}{T}} + \frac{(\Delta-1) t_{j_1 1}}{\frac{t_j}{T}}$.

When the j^{th} processed job of SPT sequence is identified, $\frac{T}{\Delta} - \sqrt{\frac{T}{T-\sum_1^{j-1} t_{j_n}}} + \frac{(\Delta-1) t_{j_1 1}}{\frac{t_j}{T}}$ is a constant. Let g denote $\frac{t_j}{T}$. With **Proposition 1**, $f(g)$ is an increasing convex function (proof shown in the Appendix) of g for $g > 0$. Thus,

$Z = f(g) + constant.$

$Zs_j = f(\frac{t_j}{T}) + constant.$

$Zs_{j+1} = f(\frac{t_{j+1}}{T}) + constant.$

If $Z(s_j) - Z(SPT) > 0$, then $Z(s_{j+1}) - Z(SPT) > 0$.

Proposition 2 has been proven.

4.3 Dynamic optimisation algorithm

The complete dynamic optimisation algorithm can be found in **Algorithm 2** (shown in Table 4).

Table 4: Algorithm of the dynamic optimisation algorithm

Algorithm 2. Dynamic optimisation algorithm
START
1. Generate a SPT sequence.
2. Enumerate job sequences based on **Proposition 1**, get S.
3. Apply Binary search method to find the job sequence s_j which satisfies $Z(s_j) < Z(\text{SPT})$ and $Z(s_{j+1}) > Z(\text{SPT})$.Let $S_j = \{s_1, s_2, ..., s_j\}$.
4. Obtain the corresponding maintenance decision for S_j based on **Lemma 1** and **Lemma 2**.
5. Apply **Algorithm 1** to calculate the makespan of each job sequence s in S_j.
6. Select the job sequence with the minimum Z as the optimal job sequence s^*.
7. Return s^*.
END

Because of the Binary search method, the dynamic optimisation algorithm has a complexity of $O(logN)$.

5 Computational experiments

Let us present the computational results and analyses. The dynamic optimisation algorithm is implemented in MATLAB R2017b on a personal computer with an Intel® Core$^{\text{TM}}$ i7-7700HQ(2.8GHZ) CPU and 8GB RAM memory under Windows 10 operating system.

5.1 Generation of instances

Data from a real valve workshop in *China Astronautics* were used to evaluate the performance of the proposed dynamic optimisation algorithm. The machine performing the process of milling the valve housing parts was chosen as the research subject in our experiments.

The analysis of the job processing times showed that their deviation has a changing trajectory. For example, Figure 3 shows the deviation of the processing time of each job in the job sequence for an instance. In every instance, there are a certain number of points that deviate from the trajectory.

Based on the analysis of 39 sets of actual processing data on the milling machine, it was established that the machine has two conditions. The transition probability between

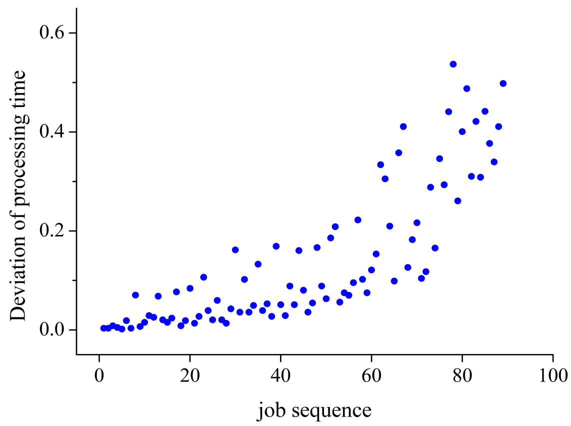


Figure 3: Deviation of job processing times in an instance

the machine conditions can be estimated by a Hidden Markov Model (HMM). A HMM is typically defined by a triplet $\lambda = \{\pi, A, K\}$:

- $\pi = \{\pi_m | m = 1, 2\}$, where π_m is the probability of the machine being in state m before starting production,
- $A = \{a_{mc} | 1 \leq m, c \leq 2\}$, where a_{mc} is the transition probability from state m to state c,
- $K = \{k_m(o_j) | m = 1, 2\}$, where $k_m(o_j)$ is the emission probability of o_j being observed at state m.

After collecting the actual job processing times, we could obtain o_j, the deviation of the actual processing time t_{ij} from its nominal value t_i. Then, the observation sequence of HMM is $O = \{o_1, o_2, ..., o_N\}$. In a HMM, given observation sequences, the transition probabilities a_{mc} can be calculated by the Baum-Welch algorithm (Baum et al., 1970).

To implement the Baum-Welch algorithm, forward variables $\alpha_j(m)$ and backward variables $\beta_j(m)$ are defined. Forward variables $\alpha_j(m)$ correspond to the joint probability of the following two events at position j: Subsequence $\{o_1, o_2, ..., o_j\}$ is observed and the machine is in state m. Thus, $\alpha_j(m) = P(o_1, o_2, ..., o_j, m|\lambda) = \left[\sum_{c=1}^{2} \alpha_{j-1}(c) a_{cm}\right] k_m(o_j)$, $j = 2, 3, ..., N$. Backward variables $\beta_j(m)$ correspond to the joint probability of the following two events at position j: Subsequence $\{o_{j+1}, o_{j+2}, ..., o_N\}$ is observed and the machine is in state m. Thus, $\beta_j(m) = P(o_{j+1}, o_{j+2}, ..., o_N, m|\lambda) = \sum_{c=1}^{2} a_{mc} k_c(o_{j+1}) \beta_{j+1}(c)$, $j = 1, 2, ..., N$.

As described in Algorithm 3, the Baum-Welch algorithm is implemented in a recursive way to estimate the transition probabilities A of the HMM.

Out of the 39 sets of data, 34 sets were used as the training data to the HMM. The rest five sets of data were used as the validation sets. After application of Baum-Welch algorithm, the HMM obtained from the training sets is applied to the validation sets and the average error is less than 5%. The transition probabilities of machine state are set as $p(0|0) = 0.78$, $p(1|0) = 0.22$, $p(1|1) = 0.60$, $p(0|1) = 0.40$. The transition probabilities after an MA are set as $p_m(0|0) = 0.85$, $p_m(1|0) = 0.15$, $p_m(1|1) = 0.10$, $p_m(0|1) = 0.90$.

Table 5: Baum-Welch algorithm for calculating transition probability

Algorithm 3. Baum-Welch algorithm

START

1. Input observation sequence $O = \{o_1, o_2, ..., o_N\}$.
2. Randomly generate initial value of parameters: $a_{mc}^{(0)} \in (0,1)$, $k_m(o_j)^{(0)} \in (0,1)$, $\pi_m^{(0)} \in (0,1)$.
3. Set $l = 0$
4. **Do**
5. $l = l + 1$
6. Calculate forward variables $\alpha_j(m)$ and backward variables $\beta_j(m)$, $j = 1, 2, ..., N$
7. Estimate values of $a_{mc}^{(l)}$, $k_m(o_j)^{(l)}$, $\pi_m^{(l)}$ using $a_{mc} = \frac{\sum_{j=1}^{N-1} \alpha_j(m) a_{mc} k_c(o_{j+1}) \beta_{j+1}(c)}{\sum_{j=1}^{N-1} \alpha_j(m)\beta_j(m)}$,
   $k_m(h) = \frac{\sum_{j=1, o_j = v_h}^{N} \alpha_j(m)\beta_j(m)}{\sum_{j=1}^{N} \alpha_j(m)\beta_j(m)}$, $\pi_m = \frac{\alpha_1(m)\beta_1(m)}{\sum_{c=1}^{2} \alpha_1(c)\beta_1(c)}$
8. **Loop Until** the values of $a_{mc}^{(l)}$, $k_m(o_j)^{(l)}$ and $\pi_m^{(l)}$ converge
9. Output the value of transition probability a_{mc}

END

Different instances were randomly generated based on the information we obtained from the above workshop. Parameters of the instances are described below.

1) The number of jobs in a batch ranges from 20 to 100.

2) There are different types of products to be processed. The processing times were generated from a normal distribution with mean value $\mu = 150$ and standard deviation $\sigma = 20$ (unit:min).

3) Other parameters are defined as follows: Maintenance time $b = 12min$; deteriorating rate $d = 0.2$.

5.2 Algorithm performance evaluation

Firstly, the performance of the dynamic optimisation algorithm is analyzed by solving real-size problems. There are nine sets with twenty randomly generated instances in each set. Table 5 reports the optimal makespan and the average times of maintenance for different instances.

Table 6: Performance of the algorithm for instances with different number of jobs.

Set No.	N	CPU(s)	Makespan(min)	Average times of MAs
1	20	0.02	5224	1.67
2	30	0.03	8116.6	2.71
3	40	0.05	10954	3.62
4	50	0.09	13928	4.72
5	60	0.12	16904	5.65
6	70	0.20	19803	6.60
7	80	0.31	22826	7.66
8	90	0.30	25841	8.60
9	100	0.44	28799	9.69

Table 6 shows that optimal solutions can be obtained within a second for instances up to 100 jobs. As expected, the times of MAs increases with the number of jobs. To evaluate our dynamic algorithm for solving real problems, we compare its performance with two maintenance strategies that are usually applied in the workshop. They are:

1) No MAs (NOMA): Maintenance is not considered in this strategy. The solution is obtained by solving the dynamic algorithm with $b = 0$;
2) MAs with Periodic Intervals (MAPI): An MA is systematically inserted into the production sequence after a given number of jobs. The production sequence is obtained by inserting MAs with periodic intervals in an optimal sequence of NOMA.

The same sets of instances were solved by the three algorithms respectively. The objective function value obtained from our dynamic algorithm is used as the benchmark (denoted as Z) for the other two strategies. The deviations are calculated as:

$$Dev_{NOMA} = \frac{Z_{NOMA} - Z}{Z} \times 100\%,\ Dev_{MAPI} = \frac{Z_{MAPI} - Z}{Z} \times 100\%.$$

where Z_{NOMA} and Z_{MAPI} are objective function values of no MAs strategy and MAs with periodic intervals strategy respectively.

Table 7 summarizes the average results for the nine sets of instances (where $Z^{(3)}_{MAPI}$, resp. $Z^{(10)}_{MAPI}$, is the objective function of MAPI with an MA inserted every three jobs, resp. every ten jobs, and $Dev^{(3)}_{MAPI}$ and $Dev^{(10)}_{MAPI}$ are the corresponding deviations). Both the average objective function values and the times of MAs are compared.

Table 7: Comparison with different maintenance strategies

		Dynamic algorithm		NOMA		MAPI				Deviation		
Set No.	N	Z (min)	Times of MAs	Z_{NOMA} (min)	Times of MAs	$Z^{(3)}_{MAPI}$ (min)	Times of MAs	$Z^{(10)}_{MAPI}$ (min)	Times of MAs	Dev_{NOMA}(%)	$Dev^{(3)}_{MAPI}$(%)	$Dev^{(10)}_{MAPI}$(%)
1	20	5224	2	5245.1	0	5291.6	6	5238.9	2	0.40	1.29	0.29
2	30	8116.6	3	8154.1	0	8213.8	10	8145.2	3	0.46	1.20	0.35
3	40	10954	4	11010	0	11109	13	10990	4	0.51	1.42	0.33
4	50	13928	5	13996	0	14111	16	13979	5	0.49	1.31	0.37
5	60	16904	6	16994	0	17155	20	16977	6	0.53	1.48	0.43
6	70	19803	7	19913	0	20107	23	19902	7	0.56	1.54	0.50
7	80	22826	8	22971	0	23194	26	22954	8	0.64	1.61	0.56
8	90	25841	9	26004	0	26248	30	25989	9	0.63	1.58	0.57
9	100	28799	10	29013	0	29308	33	28981	10	0.74	1.77	0.63

The following observations can be made:

1) Comparing with the NOMA strategy, our dynamic algorithm performs better in terms of makespan even though conducting maintenance actions takes extra time. The average deviation increases steadily with the number of jobs (from 0.40% to 0.74%). The reason is that by conducting maintenance actions during production, the probability of the machine staying at the stable state is higher. Thus, the expected processing times of the jobs after the MA are shorter. In other words, the processing efficiency of the machine is improved.
2) Our dynamic algorithm also outperforms the MAPI strategy in terms of makespan. Even if there are MAs at the same times than the dynamic algorithm and MAPI

with every ten jobs in each set, the average deviation is between 0.29% and 0.63%.

To further analyze the impact of different maintenance strategies on makespan, ANOVA test is conducted to evaluate the performance of the proposed dynamic algorithm in a statistical sense. ANOVA test is an analysis of variance for significance test. The test result may reject or support the idea that dependent variables are influenced by independent variables. In ANOVA test, the null hypothesis is that all group means are exactly equal, and the alternative hypothesis is that not all group means are equal.

Table 8: One-Way ANOVA test

	Sum of Squares	Degrees of Freedom	Mean Square	F	P-value	$F_{0.05}$-distribution
Between Groups	736 229	2	368 114.6	3.482225	0.037427	3.158843
Within Groups	6 025 610	57	105 712.5	N/A	N/A	N/A
Total	6 761 839	59	N/A	N/A	N/A	N/A

In Table 8, F is the ANOVA test statistic that shows how likely the makespan means of three algorithms are equal.

It can be observed in Table 8 that $F > F_a$ and P-value < 0.05. It indicates that the null hypothesis is rejected, in favor of alternative hypothesis. Therefore, at a significance level of 0.05, the proposed dynamic algorithm has statistical significance on makespan compared with NOMA and MAPI.

In the following, the average deviation of the objective functions obtained with five typical intervals for MAPI are compared to the optimal solution obtained from the dynamic algorithm. Figure 4 shows this average deviation for all instances for MAPI with maintenance actions inserted every three jobs, five jobs, ten jobs, fifteen jobs and twenty jobs.

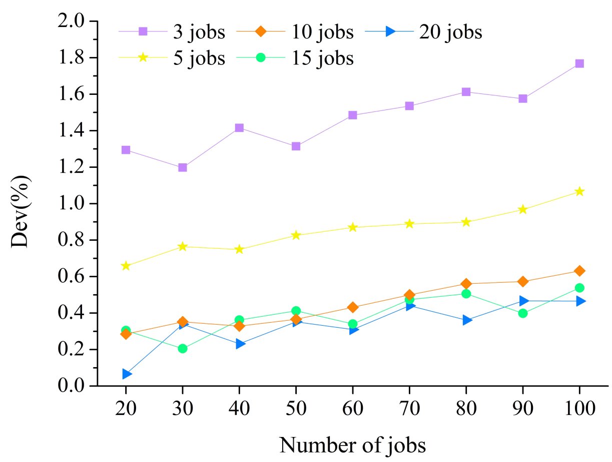


Figure 4: Comparison of different maintenance strategies

Figure 4 shows that:

1) The deviations of the MAPI strategies over the dynamic algorithm increase with

the number of jobs. For example, the deviation of the MAPI strategy with MAs inserted every five jobs increases from 0.66% to 1.07%.

2) While conducting maintenance improves the processing efficiency of the machine, more frequent maintenance actions induce an increase of the makespan. The same results are observed for different instance sizes. As the interval of MAPI increases from 3 to 20 jobs, the deviation significantly decreases in each set. For example, for an instance with 80 jobs, the deviation decreases from 1.61% to 0.36% when the interval of MAPI increases from 3 to 20 jobs.

The managerial insight of the above analysis for the workshop is to determine the appropriate maintenance intervals such that the balance between keeping processing efficiency of the machine and improving machine throughput can be achieved.

5.3 Sensitivity Analyses

Sensitivity analyses were also conducted by varying the following two parameters: (1) the transition probability of the machine state, and (2) the deteriorating rate of the processing times.

5.3.1 Transition probability of machine state

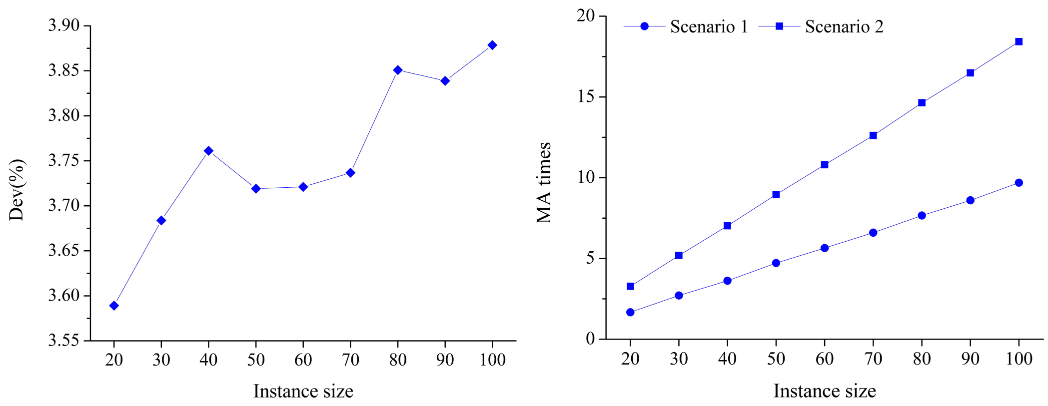
Two scenarios with different transition probabilities are considered and details are given in Table 2.

In scenario 1, the transition probability satisfies $p(0|0) + p(1|1) > 1$ and $p_m(0|0) + p_m(1|1) \leq 1$. In this case, the machine has a higher transition probability to stay in a stable state. In scenario 2, the transition probability satisfies $p(0|0) + p(1|1) > 1$ and $p_m(0|0) + p_m(1|1) > 1$. The machine has a lower transition probability to stay in a stable state.

Figure 5(a) shows the average deviation of the makespan under the two different scenarios and is calculated as:

$$Dev = \frac{Z_2 - Z_1}{Z_1} \times 100\%,$$

where Z_1 and Z_2 are the makespan for scenario 1 and scenario 2 respectively. Figure 5(b) shows the average times of MAs.



(a) Deviation of makespan (b) Average MAs times

Figure 5: Sensitivity analysis to transition probability

Figure 5(a) shows that, as the number of jobs increases, the deviation of makespan gradually rises from 3.59% to 3.88%. The reason is that more MAs are needed in scenario 2.

Figure 5(b) shows that much more maintenance actions are needed in scenario 2. For example, for an instance with 90 jobs, there are more than 8 MAs. That is, an MA is needed before the job is processed for almost one fifth of the jobs. In this case, a major preventive maintenance is strongly recommended.

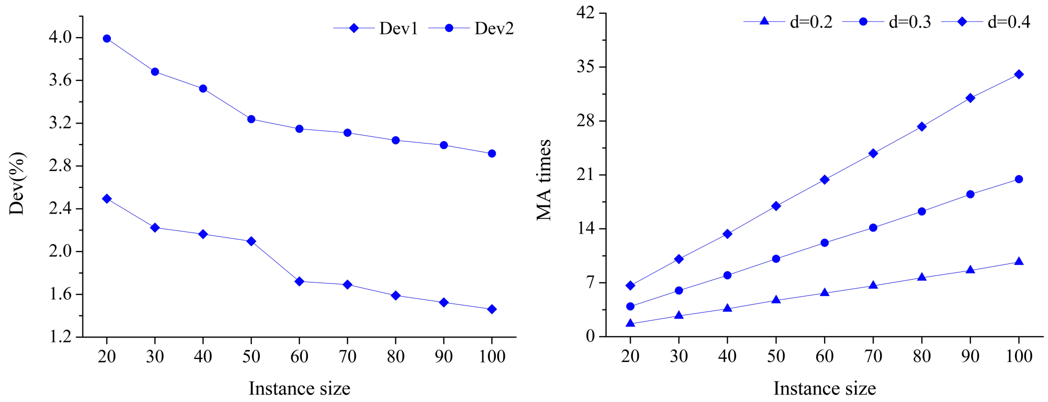
5.3.2 Deteriorating rate of processing times

For different machines, the deteriorating rate of processing times is different. To evaluate its impact on makespan, three levels of deteriorating rates are considered in this set of experiments, namely low deteriorating rate ($d = 0.2$), medium deteriorating rate ($d = 0.3$), and high deteriorating rate ($d = 0.4$).

Figure 6(a) illustrates the average deviation of makespan with three different deteriorating rates. Figure 6(b) illustrates the average times of MAs. The makespan with $d = 0.2$ is used as the benchmark to calculate the deviations:

$$Dev_1 = \frac{Z^{d_2} - Z^{d_1}}{Z^{d_1}} \times 100\%, \; Dev_2 = \frac{Z^{d_3} - Z^{d_1}}{Z^{d_1}} \times 100\%$$

where Z^{d_1}, Z^{d_2} and Z^{d_3} are the makespan for low deteriorating rate, medium deteriorating rate and high deteriorating rate respectively.



(a) Deviation of makespan (b) Average MAs times

Figure 6: Sensitivity analysis to different deteriorating rates

It can be observed in Figure 6 that:

1) With higher deteriorating rate (d), the average deviation of makespan is larger. This tendency becomes less apparent for larger size problems. When d increases from 0.2 to 0.3, the deviation gradually declines from 2.49% to 1.46%. When d increases from 0.2 to 0.4, the deviation declines from 3.99% to 2.92%. It reveals that the deteriorating rate has a less significant impact on the processing time as the number of jobs increases.

2) As d increases, more MAs are needed. For instances with 90 jobs, when d increases from 0.2 to 0.3, ten more MAs are needed. And when d increases from 0.2 to 0.4, twenty-three more MAs are needed. It is obvious that the increasing of deteriorating rate leads to more MAs to maintain processing efficiency as the number of jobs increases.

In this study, we consider that the machine is at the middle stage of the bathtub curve, and its reliability is at a stable level. Even so, the machine condition still has an impact on processing efficiency. Our dynamic optimisation approach provides optimal schedules by modifying the SPT sequence and inserting minor maintenance activities. Since the transition probability of the machine condition to stay in a state stable is higher after an MA, performing more maintenance actions help to maintain processing efficiency. Thus, the machine is kept in a more stable state, and the proposed schedule becomes more efficient for practical use. From the experimental results, if the times of minor maintenance actions are large, then a major preventive maintenance action is recommended to restore machine condition.

6 Conclusion

We have studied a single machine scheduling problem with machine conditions and maintenance decisions derived from a valve manufacturing company. The processing time of a job is assumed to be dependent on its position in the schedule and the machine conditions. Discrete machine states are defined via transition probabilities. An

MDP model is formulated to minimize makespan. Optimality properties are analyzed, based on which a dynamic optimisation algorithm is developed for solving real-world problems by modifying the SPT sequence. Computational experiments demonstrate the effectiveness and the efficiency of our dynamic algorithm to minimize makespan with machine condition constraints. The performance of the dynamic algorithm is further validated by comparing with two maintenance strategies. It is revealed that the makespan can be reduced by 0.50% averagely by joint optimisation of production sequence and maintenance decisions.

Sensitivity analyses provide valuable information on the impact of the state transition probability and the deteriorating rate on the optimized schedules. When the transition probability of the machine staying in a stable state decreases, more maintenance actions are needed to minimize the makespan. When the machine deteriorating rate increases, the times of the maintenance actions increase. The managerial implication is twofold: 1) Considering machine conditions helps to maintain the processing efficiency of the machine and makes the production schedule more practical; and 2) The computational results help the workshop managers to make appropriate maintenance decisions to improve the throughput of the machine.

An interesting extension of this research is to study the problem in a more complex production environment, such as parallel machine scheduling and flow shop scheduling problems where each machine may have different transition probabilities to indicate machine conditions. The algorithm for the single machine problem developed in this study could be embedded into a decomposition approach to quickly obtain good solutions to be used in more complex systems.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Project number: 51775347). We thank the referees for their valuable comments.

References

[1] Allahverdi, A. and Mittenthal, J. 1994. Scheduling on m parallel machines subject to random breakdowns to minimize expected mean flow time. Naval Research Logistics, 41: 677–682.

[2] Allahverdi, A. and Mittenthal, J. 1995. Scheduling on a two-machine flowshop subject to random breakdowns with a makespan objective function. European Journal of Operational Research, 81: 376–387.

[3] Ahmadzadeh, F. and Lundberg, J. 2013. Remaining useful life prediction of grinding mill liners using an artificial neural network. Minerals Engineering, 53: 1–8.

[4] Alsina, E. F., Chica, M., Krzysztof T. and Regattieri, A. 2017.On the use of machine learning methods to predict component reliability from data-driven indus-

trial case studies. International Journal of Advanced Manufacturing Technology, 94: 2419–2433.

[5] Baum, L. E., Petrie, T., Soules, G. and Weiss, N. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. The Annals of Mathematical Statistics, 41(1): 164-171.

[6] Boukas, E. K. and Liu, Z. K. 2001. Production and maintenance control for manufacturing systems. IEEE Transactions on Automatic Control, 46: 1455–1460.

[7] Boudjelida, A. 2019. On the robustness of joint production and maintenance scheduling in presence of uncertainties. Journal of Intelligent Manufacturing, 30: 1515–1530.

[8] Broea, M. A.J., Teunter, R. H., Jonge, B., Veldman, J., Foreest, N. D. 2020. Condition-based production planning: Adjusting production rates to balance output and failure risk. Manufacturing & Service Operations Management, 22(4): 792-811.

[9] Cassady, C.R. and Kutanoglu, E. 2005. Integrating preventive maintenance planning and production scheduling for a single machine. IEEE Transactions on Reliability, 54(2): 304–309.

[10] Cheng, T.C.E.a, Lee, W.C. and Wu, C.C. 2010. Single-machine scheduling with deteriorating functions for job processing times. Applied Mathematical Modelling, 34: 4171–4178.

[11] Castanier, B., Gehan, M. and Lemoine, D. 2015. Joint optimisation of a master production schedule and a preventive maintenance policy. International Conference on Industrial Engineering & Systems Management, 1–7.

[12] Cheng, G. Q., Zhou, B. H., Li, L. 2016. Joint optimisation of production rate and preventive maintenance in machining systems. International Journal of Production Research, 54(21): 6378-6394.

[13] Dieulle, L., Berenguer,C., Grall, A. and Roussignol, M. 2003. Sequential condition-based maintenance scheduling for a deteriorating system. European Journal of Operational Research, 150: 451–461.

[14] Gorjian, N., Ma, L., Mittinty, M., Yarlagadda,P. and Sun, Y. 2010. A review on degradation models in reliability analysis. In: Kiritsis D., Emmanouilidis C., Koronios A., Mathew J. (eds) Engineering Asset Lifecycle Management. Springer, London 369–384.

[15] Gonzalez, P. P. and Framina, J.M. 2018. Single machine scheduling with periodic machine availability. Computers & Industrial Engineering, 123: 180–188.

[16] Han, S., Jin, X. N. and Li, J. X. 2016. An assessment method for the impact of missing data in the rough set-based decision fusion. Intelligent Data Analysis, 20: 1267–1284.

[17] Huang, Z.W., Zhu, J. M., Lei, J. T., Li, X. R., and Tian, F. Q. 2019. Tool wear predicting based on multi-domain feature fusion by deep convolutional neural network in milling operations. Journal of Intelligent Manufacturing, 31: 953-966.

[18] Johnson, B. W. 1989.The Design and Analysis of Fault Tolerant Digital Systems. USA: Addison-Wesley.

[19] Ji, M., He, Y. and Cheng, T. C. E. 2007. Single-machine scheduling with periodic maintenance to minimize makespan. Computers & Operations Research, 34: 1764–1770.

[20] Kurt M. and Kharoufeh, J. P. 2009. Monotone optimal replacement policies for a Markovian deteriorating system in a controllable environment. Operation Research Letters, 38: 273–279.

[21] Kim, M.J., Jiang, R., Makis, V. and Lee, C.G. 2011. Optimal Bayesian fault prediction scheme for a partially observable system subject to random failure. European Journal of Operational Research, 214: 331–339.

[22] Khaleghei A. and Makis V. 2015. Model parameter estimation and residual life prediction for a partially observable failing system. Naval Research Logistics, 62: 190-205.

[23] Kao, Y.-T., Dauzère-Pérès, S., Blue, J., and Chang, S.-C. 2018. Equipment Health Integrated Scheduling for Semiconductor Fabrication. Computers & Industrial Engineering, 120: 450–459.

[24] Lee, C. Y. 1996. Machine scheduling with an availability constraint. Journal of Global Optimization, 9: 395–416.

[25] Lee, C. Y. 1997. Minimizing the makespan in the two machine flowshop scheduling problem with an availability constraint.Operations Research Letters, 20: 129–139.

[26] Low, C., Hsu, C. J. and Su, C. T. 2010. A modified particle swarm optimization algorithm for a single-machine scheduling problem with periodic maintenance.Expert Systems with Applications, 34: 6429–6434.

[27] Lee, J., Lapira, E., Bagheri, B. and Kao, H. 2013. Recent advances and trends in predictive manufacturing systems in big data environment. Manufacturing Letters, 1: 38–41.

[28] Lee, J. Y. and Kim, Y. D. 2015. A branch and bound algorithm to minimize total tardiness of jobs in a two identical-parallel-machine scheduling problem with a machine availability constrain. Journal of the Operational Research Society, 66: 1542–1554.

[29] Li, Y., Mohan, K., Sun, H. and Jin, R. 2017. Ensemble modeling of in situ features for printed electronics manufacturing with in situ process control potential. IEEE Robotics and Automation Letters, 2: 1864–1870.

[30] Liu, Q., Dong, M., Chen, F. F., Lv, W. and Ye, C. 2018. Single-machine-based joint optimization of predictive maintenance planning and production scheduling. Robotics and Computer-Integrated Manufacturing, 51: 238–247.

[31] Makis, V., Jiang, X. and Jardine, K. S. 1998. A condition-based maintenance model. IMA Journal of Mathematics Applied in Business & Industry, 9: 201–210.

[32] Martinelli F. 2005. Control of manufacturing systems with a two-value, production-dependent failure rate. Automatica, 41(11): 1943-1948.

[33] Mokhtari, H., Mozdgir, A. and Abadi, I. N. 2012. An integrated production and maintenance planning model with time windows and shortage cost. International Journal of Production Research, 50: 5906–5925.

[34] Nourelfath, M., Fitouhi, M.C. and Machani, M. 2010. An integrated model for production and preventive maintenance planning in multi-state systems. IEEE Transactions on Reliability, 59: 496–506.

[35] Najid, N. M., Alaoui-Selsouli, M. and Mohafid, A. 2011. An integrated production and maintenance planning model with time windows and shortage cost. International Journal of Production Research, 49: 2265–2283.

[36] Neves, M.L., Santiago, L. P. and Maia, C.A. 2011. A condition-based maintenance policy and input parameters estimation for deteriorating systems under periodic inspection. Computers & Industrial Engineering, 61: 503–511.

[37] Nesello, V., Subramainian, A., Bsttarra, M. and Laporte, G. 2018. Exact solution of the single-machine scheduling problem with periodic maintenances and sequence-dependent setup times. European Journal of Operational Research, 266: 498–507.

[38] Peng, H. and van Houtum, G. J. 2016. Joint optimization of condition-based maintenance and production lot-sizing. European Journal of Operational Research, 253: 94–107.

[39] Regattieri, A., Manzini, R. and Battini, D. 2010. Estimating reliability characteristics in the presence of censored data: a case study in a light commercial vehicle manufacturing system. Reliability Engineer System Safety, 95: 1093–1102.

[40] Sloan T W. 2004. A periodic review production and maintenance model with random demand, deteriorating equipment, and binomial yield. Journal of The Operational Research Society, 55(6): 647-656.

[41] Sun, H., Rao, P. K., Kong, Z. J., Deng, X. and Jin, R. 2017. Functional quantitative and qualitative models for quality modeling in a fused deposition modeling process.IEEE Transactions on Automation Science and Engineering, 15: 393–403.

[42] Shui, H.Y., Jin, X. N. and Ni, J. 2019. Twofold variation propagation modeling and analysis for roll-to-roll manufacturing systems. IEEE Transactions on Automation Science and Engineering, 16: 599–612.

[43] Tamssaouet, K., Dauzère-Pérès, S., and Yugma, C. 2018. Metaheuristics for the Job-Shop Scheduling Problem with Machine Availability Constraints. Computer & Industrial Engineering, 125: 1–8.

[44] Tian, W., Meina, Q., Mengyi, Z., Yi, Y., and Hichem, S. 2018. Data-driven prognostic method based on self-supervised learning approaches for fault detection. Journal of Intelligent Manufacturing, 31:1611-1619.

[45] Ting, W., Roberto, B., Andrew, L. and Qian, H. 2018. A branch-and-price algorithm for scheduling of deteriorating jobs and flexible periodic maintenance on a single machine. European Journal of Operational Research, 271: 826–838.

[46] Xie, M., Tang, Y., and Goh, T.N. 2002. A modified Weibull extension with bathtub-shaped failure rate function. Reliability Engineering & System Safety, 76: 279–285.

[47] Yan, J. and Hua, D. 2010. Energy consumption modeling for machine tools after preventive maintenance. 2010 IEEE International Conference on Industrial Engineering & Engineering Management, China, 2201–2205.

[48] Yang, S. J. 2013.Unrelated parallel-machine scheduling with deterioration effects and deteriorating multi-maintenance activities for minimizing the total completion time. Applied Mathematical Modelling, 37: 2995–3005.

[49] Yildirim, M. B. and Nezami, F. G. 2014. Integrated maintenance and production planning with energy consumption and minimal repair. The International Journal of Advanced Manufacturing Technology, 74: 1419–1430.

[50] Yu, X., Zhang, Y. and Steiner, G. 2014.Single-machine scheduling with periodic maintenance to minimize makespan revisited. Journal of Scheduling, 17: 263–270.

[51] Zhou, J. L., Cao, K., Cong P.J., Wei, T. Q., Chen, M. S., Zhang, G. X., Yan, J. M. and Ma, Y. 2017. Reliability and temperature constrained task scheduling for makespan minimization on heterogeneous multi-core platforms. The Journal of Systems and Software, 133: 1–16.

Appendix

1. Proof of Lemma 3.

Proof:

1) There is no MA between position $j-1$ and position j.

From (4) and (6), the difference between $C_j^{'}$ and $C_j^{''}$ is,

$$\begin{aligned}\Delta C &= C_j^{'} - C_j^{''} \\ &= p(0|0)t_{ij} + (1-p(0|0))t_{ij}^{'} - (1-p(1|1))t_{ij} - p(1|1)t_{ij}^{'} \\ &= (1-p(0|0)-p(1|1))t_{ij}.\end{aligned}$$

It can be observed that $t_{ij} \geq 0$, $p(0|0)+p(1|1) > 1$, $\Delta C \leq 0$.

2) An MA is inserted between position $j-1$ and position j.

Let C_j^{MA} be the expected completion time at position j ($j = 1, 2, ..., N$) if the machine is in a stable state at the beginning of position j; and $C_j^{MA^{'}}$ be the expected completion time at position j ($j = 1, 2, ..., N$) if the machine is in an unstable state at the beginning of position j. From (5) and (7), the difference between C_j^{MA} and $C_j^{MA^{'}}$ is,

$$\begin{aligned}\Delta C^{MA} &= C_j^{MA} - C_j^{MA^{'}} \\ &= b + p_m(0|0)t_{ij} + (1-p_m(0|0))t_{ij}^{'} - b - (1-p_m(1|1))t_{ij} - p_m(1|1)t_{ij}^{'} \\ &= (1-p_m(0|0)-p_m(1|1))t_{ij}.\end{aligned}$$

It can be observed that $t_{ij} \geq 0$, $p_m(0|0)+p_m(1|1) > 1$, $\Delta C^{MA} \leq 0$.

Lemma 3 has been proven.

2. Proof of Proposition 2.

In proposition 2, $f(g) = \frac{(c-g)^{-\frac{1}{2}} - c^{-\frac{1}{2}}}{g}$ is an increasing convex function.

Proof:

Le us assume that $f(g) = \frac{(c-g)^{-\frac{1}{2}} - c^{-\frac{1}{2}}}{g}$ is an increasing convex function, and so is $-f(c-g) = \frac{g^{-\frac{1}{2}} - c^{-\frac{1}{2}}}{g-c}$. Denote $g_1 > g_2 > 0$, $f(g_1) - f(g_2) = f(c-g_2) - f(c-g_1) = \frac{g_1^{-\frac{1}{2}} - c^{-\frac{1}{2}}}{g_1 - c} - \frac{g_2^{-\frac{1}{2}} - c^{-\frac{1}{2}}}{g_2 - c}$.

With $(g_1 - c)(g_2 - c) > 0$, the fractional molecule of $f(c-g_2) - f(c-g_1)$ can be written as:

$$g_1^{-\frac{1}{2}} g_2 - g_2^{-\frac{1}{2}} g_1 + c^{-\frac{1}{2}}(g_1 - g_2) - c(g_1^{-\frac{1}{2}} - g_2^{-\frac{1}{2}})$$

which can be rewritten,

$$g_1^{-\frac{1}{2}} g_2^{-\frac{1}{2}} (g_2^{-\frac{3}{2}} - g_1^{-\frac{3}{2}}) + c^{-\frac{1}{2}}(g_1 - g_2) - c(g_1^{-\frac{1}{2}} - g_2^{-\frac{1}{2}})$$

where $g_1 > g_2 > 0$, then $g_2^{-\frac{3}{2}} - g_1^{-\frac{3}{2}} > 0$, $c^{-\frac{1}{2}}(g_1 - g_2) > 0$, and $g_1^{-\frac{1}{2}} - g_2^{-\frac{1}{2}} < 0$.

According to the properties of increasing functions, $g_1 > g_2 > 0$ and $f(c-g_2) - f(c-g_1) > 0$, $-f(c-g)$ is an increasing function. Thus, the function $f(g) = \frac{(c-g)^{-\frac{1}{2}} - c^{-\frac{1}{2}}}{g}$ is an increasing convex function.

The proposition has been proven.

3. Deviation between MAPI and NOMA with different job intervals.

Figure 7 presents the deviation between MAPI and NOMA with job intervals for MAPI from 1 to 100.

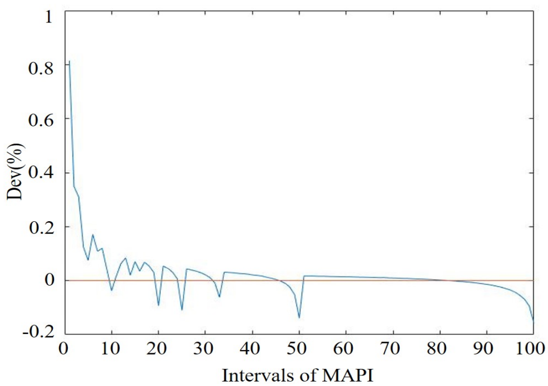


Figure 7: Comparison of MAPI and NOMA

Figure 7 shows that, with the increase of the job interval, the makespan obtained with MAPI approaches the makespan obtained with NOMA. When the job interval for MAPI is lower than 45 jobs, the deviation decreases from 0.8% to 0 in a fluctuating way. Except for some job intervals, too many MAs make the makespan worse. When the job interval is between 45 and 50 jobs, the deviation is below 0. When there are not too many MAs, MAPI improves the machine condition and reduces the makespan. When the job interval is greater than 50, there is only one MA in MAPI at a different job position. The later the MA is inserted from the 51^{th} job position, the more significant the effect of the maintenance action.