Scheduling problems with rejection and piece-rate maintenance to minimise the total weighted completion time

Xianyu Yu and Zhen Wang

College of Economics and Management,
Nanjing University of Aeronautics and Astronautics,
29 Jiangjun Avenue, Nanjing 211106, China
Email: xyyu@nuaa.edu.cn
Email: wzhen@nuaa.edu.cn

Kai Huang*

DeGroote School of Business,
McMaster University,
Hamilton, L8S4M4, Canada
Email: khuang@mcmaster.ca
*Corresponding author

Dehua Xu

School of International Economics and Business,
Nanjing University of Finance & Economics,
3 Wenyuan Avenue, Nanjing 210023, China
Email: xudehua@nufe.edu.cn

Xiuzhi Sang

College of Economics and Management,
Nanjing University of Aeronautics and Astronautics,
29 Jiangjun Avenue, Nanjing 211106, China
Email: sangxz@nuaa.edu.cn

Abstract: This paper addresses the single machine scheduling problems with simultaneous consideration of rejection and piece-rate maintenance. Each job is either accepted to be processed on the machine, or rejected in which case a rejection penalty will be incurred. The piece-rate maintenance refers that the machine performs maintenance activity every time it completes a given number of jobs. The objective is to minimise the sum of weighted completion times, rejection costs and maintenance costs. Our contribution is threefold. First, the general case of the considered problem is proved to be NP-hard, and an approximate algorithm is developed to solve the problem. Second, for the case with agreeable condition that jobs with smaller processing times are weighted more, a pseudo-polynomial algorithm is developed to establish that the problem is NP-hard only in the ordinary sense. This pseudo-polynomial algorithm is further converted into a fully polynomial time approximation scheme (FPTAS).

In the third, two special cases, in which one with all equal weights and the other one with all equal processing times, are proved to be solved in polynomial time.

Keywords: scheduling; rejection; maintenance; agreeable condition; fully polynomial time approximation scheme; FPTAS.

Reference to this paper should be made as follows: Yu, X., Wang, Z., Huang, K., Xu, D. and Sang, X. (2022) 'Scheduling problems with rejection and piece-rate maintenance to minimise the total weighted completion time', *Int. J. Automation and Control*, Vol. 16, No. 1, pp.64–86.

Biographical notes: Xianyu Yu is an Associate Professor in the Department of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He has studied a joint PhD in Management Science from Southeast University, Nanjing, China and McMaster University, Hamilton, Canada. His research interests include carbon efficient scheduling, artificial algorithms and energy system simulation.

Zhen Wang is a postgraduate in the Department of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He received his BE from Nanchang University, Nanchang, China. His research interests are in the area of carbon efficient scheduling and algorithm analysis.

Kai Huang is an Associate Professor in DeGroote School of Business at McMaster University, Hamilton, Canada. He received his PhD in Industrial Engineering from Georgia Institute of Technology, Atlanta, USA. His interests include logistics network capacity expansion, inventory management, humanitarian logistics, and food safety.

Dehua Xu is an Associate Professor in the School of International Economics and Business at Nanjing University of Finance & Economics, Nanjing, China. He received his PhD in Science from Beijing Normal University, Beijing, China. His research interests are in the areas of production scheduling, emergency management, and electronic business.

Xiuzhi Sang is an Associate Professor in the Department of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing, China. She received her PhD in Management Science from Southeast University, Nanjing, China. Her research interests include optimisation and decision analysis.

1 Introduction

In order to maximise profits, lean philosophy is widely adopted in manufacturing enterprises. One of the important areas of lean philosophy application in manufacturing enterprises is the production scheduling in workshop. In the real manufacturing process, if production scheduling, maintenance planning and product quality control are interdependent and making these decisions independently, it may lead to suboptimal solutions (Hadidi et al., 2011). In order to make workshop operation lean, production scheduling, maintenance planning, quality control and other operation strategies should be synchronised with each other (Kumar and Lad, 2017).

Among the scheduling research in this field, order rejection and machine maintenance have become two hot issues that have been widely concerned by many scholars. The research works associated with two above issues can be classified into three categories, viz., scheduling with job rejection, scheduling with machine maintenance and scheduling with the integrated consideration of machine maintenance and job rejection.

Scheduling problems with rejection have been paid a great deal of attention by researchers over the last decade (Shabtay et al., 2013). In highly loaded make-to-order manufacture systems, rejecting the processing of some jobs by either outsourcing them or just rejecting them may reduce the inventory and tardiness costs at the price of outsourcing cost or the loss in income and customer goodwill. A high level decision of splitting the jobs into accepted and rejected parts should be made prior to the scheduling decision (Wang et al., 2016). Bartal et al. (2000) was one of pioneers considering scheduling problems with job rejection, and studied the problem of minimising the sum of makespan and rejection penalty on identical parallel machines. Some other studies on scheduling with rejection concerning the makespan criterion include Lu et al. (2009), Zhang et al. (2010), Qi (2011) and Ou et al. (2015). For the criterion associated with the sum of completion times, Engels et al. (2003) presented a fully polynomial time approximation scheme (FPTAS) for the single machine scheduling problem to minimise the sum of weighted completion times and rejection costs. Some related works concerning the sum of completion times include Zhang et al. (2010) and Shabtay (2014). Several studies on scheduling with rejection concerned the tardiness criterion (Thevenin et al., 2015; Kannan et al., 2018; Somasundaram and Benjanarasuth, 2019). Several other studies considered the integrated criterions of the due-date assignment and other scheduling objectives (Steiner and Zhang, 2011; Selvarajah and Zhang, 2014; Yin et al., 2015; Mohamed et al., 2018a; Xu and Xu, 2018; Kong et al., 2019).

For the scheduling problems with machine maintenance, the interrelated approaches are adopted in many researches. The machine maintenance activities are operated regularly to keep production effectiveness and efficiency in the actual manufacturing process. Due to its practical importance, scheduling with availability constraints attracted lots of attentions from the community in the last two decades (Cheng and Wang, 1999; Gawiejnowicz, 2007; Xu et al., 2008, 2009, 2015). A piece-rate maintenance activity, first introduced to scheduling by Yu et al. (2013), Can keep the production efficiency under the piecework production workshop. Xue et al. (2014) extended the problem setting of Yu et al. (2013) by adding the investigation of interval constrained position-dependent processing times, and proposed the optimal polynomial algorithm to solve the considered problem. Recent developments associated with maintenance activities include those studied by Yin et al. (2016, 2017, 2018), Cui and Lu (2017), Guo et al. (2017), Li et al. (2020), Liu et al. (2018), Mohamed et al. (2018b), Omar and Shaik (2019) and Pati and Negi (2019).

It is natural and interesting to investigate scheduling problems with the integrated consideration of job rejection and machine maintenance. Zhang and Luo (2013) considered such problems on parallel machines with a certain time interval reserved for other usages on one of the machines. An FPTAS was presented to minimise the sum of makespan and rejection cost for the two-identical-machine case. Zhong et al. (2014) extended the work of Zhang and Luo (2013) to a more general situation that there are multiple time intervals prevented from processing jobs. Similarly, an FPTAS together with some approximation properties were presented in the paper. Wang et al. (2016) investigated the single-machine scheduling problems with simultaneous consideration of job rejection, controllable processing times and rate-modifying activity, and aimed to find the trade-off between two criteria associated with total completion time and the rejection cost. Kumar and Lad (2017) proposed a simulation-based optimisation approach to solve the parallel-machine scheduling problem with the integrated consideration of maintenance planning and the effect of cost of rejection. The considered objective is to minimised the overall operations cost associated with due-date assignment, maintenance cost and rejection cost. Wang et al. (2018) considered a rescheduling problem with the unavailable constraint and rejection cost on the parallel-machine setting.

It can be seen from the above literature review that scheduling problems with the integrated consideration of maintenance and rejection are becoming an important area of scheduling research. However, the current literatures in this field have not included the integrated investigation of job rejection and piece-rate maintenance. Further, to the best of our knowledge no attempt is made to investigate the criterion related with the total weighted completion time under the integrated investigation of machine maintenance and job rejection. In order to address the above gaps, this paper aims to investigate single-machine scheduling problems with job rejection and piece-rate maintenance, and aims to minimise the sum of weighted completion times, rejection penalties and maintenance cost.

To motivate our considered scheduling problem, consider a practical example related to the manufacturing of glass products. In this context, the production process of each product should be non-preemptive in order to prevent the glass melting liquid from solidifying into waste product. Because each glass manufacturing enterprise has its own capacity constraint, the enterprise does not always process all the jobs. However, rejecting the processing of some jobs may reduce the inventory and tardiness costs at the price of outsourcing cost or the loss in income and customer goodwill. Since the glass melting liquid is easy to adhere to the machine and waste usually occurs during the process of pouring and blowing, the machine needs to be cleaned and maintained every time a certain number of glass products are completed. Due to the temperature sensitivity and fragility of glass products, the cost of inventory and disposal of glass products has become an important part of total operation cost in the production process. This situation can be modelled as our problem of scheduling with job rejection and piece-rate maintenance of minimising the sum of weighted completion times, rejection and maintenance costs.

The rest of the paper is organised as follows. In Section 2, the considered problems are formulated. In Section 3, the dynamic programming approximation algorithm for the first problem is discussed, and then both pseudo-polynomial algorithm and FPTAS are developed to solve the second problem. Section 4 studies the third and fourth problems, where all jobs have either equal weights or equal processing times. In Section 5, we conclude the paper and suggest several future research topics.

2 Problem formulation

The considered single-machine scheduling problems with job rejection and piece-rate maintenance can be stated as follows. There is a set of n independent jobs $J = \{J_1, J_2, \ldots,$

$J_n\}$ to be scheduled on a single machine. All the jobs are available at time zero, and job preemption is not allowed during job processing on the machine. Let p_j, e_j and w_j be the processing time, rejection cost and weight of job J_j, respectively. In order to maintain processing efficiency, a piece-rate maintenance activity is executed on the machine every B jobs processed. The machine is ready for processing jobs at time zero. Let s be the time length of each maintenance activity. The cost of carrying out a maintenance activity is q. Without losing generality, this paper assumes that all the above parameters are non-negative integers.

For the scheduled job sequence σ, let $S(\sigma)$ and $\bar{S}(\sigma)$ be the set of processed and rejected jobs, respectively. It is obvious that $S(\sigma) \cap \bar{S}(\sigma) = \varnothing$ and $S(\sigma) \cup \bar{S}(\sigma) = J$. For job $J \in S(\sigma)$, let $C_j(\sigma)$ be the completion time of σ. For simplicity, S, $\bar{S}$ and C_j are used when the referred schedule σ is clear to the readers. Thus, the total cost of σ can be denoted by

$$TC = \sum_S w_j C_j + \sum_{\bar{S}} e_j + bq, \tag{1}$$

where b denotes the number of maintenance activities. It can be calculated by $b = \lceil |S| / B \rceil - 1$, where $|S|$ is the number of processed jobs.

In this paper, we consider four different optimisation problems that can arise for each problem setting and constraints.

- The first one, denoted by P1, is based on the setting of piece-rate maintenance. Using the three-field notation scheme $\alpha \mid \beta \mid \gamma$ introduced by Graham et al. (1979), the problem can also be denoted as $1 \mid PRM \mid \sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$, where PRM stands for piece-rate maintenance.
- The second one, denoted by P2, is based on the setting of piece-rate maintenance and the agreeable constraint that the weights and processing times are agreeable, i.e., for any two jobs J_i and J_j, $p_i / w_i \le p_j / w_j$ implies $w_i \ge w_j$. Based on the three-field notation scheme $\alpha \mid \beta \mid \gamma$, P2 can also be denoted as $1 \mid PRM, AWP \mid \sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$, where AWP denotes the agreeable constraint of weights and processing times.
- The third one, denoted by P3, is based on the setting of piece-rate maintenance and the constraint that all the weights are equal, i.e., $w_j = w$ for $j = 1, 2, \ldots, n$. Following the three-field notation scheme, $\alpha \mid \beta \mid \gamma$ this problem can also be denoted by $1 \mid PRM, w_j = w \mid w \sum_S C_j + \sum_{\bar{S}} e_j + bq$.
- The fourth one, denoted by P4, is based on the setting of piece-rate maintenance and the constraint that all the processing times of jobs are equal, i.e., $p_j = p$ for $j = 1, 2, \ldots, n$. Based on the three-field notation scheme $\alpha \mid \beta \mid \gamma$, we refer to this problem as $1 \mid PRM, p_j = p \mid \sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$.

3 Analysis of the P1-P2 problems

In this section, we focus on the P1-P2 problems. In what follows, this paper first analyses the computational complexity and the solving algorithm for the P1 problem, followed by developing a pseudo-polynomial algorithm for the P2 problem, and then convert the pseudo-polynomial algorithm into an FPTAS for the P2 problem.

3.1 Analysis of the P1 problem

This subsection aims to explore the Np-hard analysis and the solving algorithm for the $1\left|PRM\right|\sum_{S} w_j C_j + \sum_{\bar{S}} e_j + bq$ problem.

Theorem 1: The $1\left|PRM\right|\sum_{S} w_j C_j + \sum_{\bar{S}} e_j + bq$ problem is NP-hard.

Proof: The single-machine scheduling problem to minimize the total weighted completion times with rejection $1\|\sum_{J_j \in S} w_j C_j + \sum_{J_j \in \bar{S}} e_j$ was proved to be NP-hard in Engels et al. (2003). In the proof, the instance I (where job J_i has $p_j = w_j = a_j$, $e_j = Aa_j + a_j^2/2$ and $A = 1/2\sum_{j=1}^{n} a_j$) was reduced to a Partitioning problem (where n items were required to be divided into two sets with equal sizes $A = 1/2\sum_{j=1}^{n} a_j$ and a_j is the size of item j, where j = 1, 2, …, n). Since the $1\|\sum_{J_j \in S} w_j C_j + \sum_{J_j \in \bar{S}} e_j$ problem is a special case of the $1\left|PRM\right|\sum_{S} w_j C_j + \sum_{\bar{S}} e_j + bq$ problem when $s = 0$ and $q = 0$, it can be seen that instance I is also an instance of the $1\left|PRM\right|\sum_{S} w_j C_j + \sum_{\bar{S}} e_j + bq$ problem. Since the Partitioning problem is a classic NP hard problem, it can be obtained that $1\left|PRM\right|\sum_{S} w_j C_j + \sum_{\bar{S}} e_j + bq$ is also a NP hard problem. □

In (1), TC is the summation of machine-time cost $PC = \sum_{S} w_j C_j + bq$ and rejection cost $RC = \sum_{\bar{S}} e_j$, Let S_i be the set including the jobs scheduled between the (i – 1)th and the ith maintenance activities, where i = 1, 2, …, b + 1. In order to keep the consistence, two dummy activities (0th and (b + 1)th maintenance) are introduced. Thus, we have $S = S_1 \cup S_2 \cup \ldots \cup S_b$.

For the scheduled job sequence in set S_i(i = 1, 2, …, b + 1), it is obvious that no maintenance activities and rejection penalties are adopted in this partial job schedule, then minimising TC is equivalent with minimising $\sum_{S_i} w_j C_j$ if the other parts of schedule keep unchanged. Then, it is easy to obtain the following proposition by the standard interexchange method.

Proposition 1: Between two adjacent maintenance activities, any schedule with two adjacent jobs J_i and J_j don't satisfy $p_i/w_i \le p_j/w_j$ can be enhanced by inter-exchanging two jobs J_i and J_j.

For the case that a piece-rate maintenance activity is inserted into two adjacent jobs J_i and J_j, we can obtain the following proposition by the interchange operation of J_i and J_j.

Proposition 2: If a maintenance activity with the time length s is inserted between two adjacent jobs J_i and J_j, any schedule with two adjacent jobs J_i and J_j don't satisfy $(p_i+\mathrm{s})/w_i \le (p_j+\mathrm{s})/w_j$ can be enhanced by inter-exchanging two jobs J_i and J_j.

Based on Proposition 1 and Proposition 2, a dynamic programming algorithm is developed to find an excellent feasible solution for the $1\left|PRM\right|\sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$ problem by making either processing or rejecting decision on jobs one-by-one. Based on Proposition 1, we index all the jobs in the weighted shortest processing time first (WSPT) order such as $p_1/w_1 \le p_2/w_2 \le \cdots \le p_n/w_n$. Let (j, k, t, v) be the state representative denoting a partial schedule on the first j jobs, where t, k and v are the completion time of the last processed job, the number of processed jobs after the last maintenance activity and the total cost of (j, k, t, v), respectively.

If $j = n$, (n, t, k, v) denotes a full schedule and also a feasible solution to the $1\left|PRM\right|\sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$ problem. For $j < n$, the next unscheduled job J_{j+1} can be scheduled in one of the following four cases.

Case 1 Job J_{j+1} is rejected. State $\left(j+1, t, k, v'\right)$ is generated, where $v' = v + e_{j+1}$.

Case 2 If $k < B$, job J_{j+1} is processed. State $(j+1, t+p_{j+1}, k+1, v')$ is generated, where $v' = v + w_{j+1}(p_{j+1}+t)$.

Case 3 If $k = B$, job J_{j+1} is processed with a maintenance activity prior to it. State $(j+1, t+s+p_{j+1}, 1, v')$ is generated, where $v' = v + w_{j+1}(p_{j+1}+s+t)+q$.

Case 4 If $k = B$, process job $j + 1$ in the current B position and process job j in the first position after maintenance. State $(j+1, t+p_{j+1}+s, 1, v')$ is generated, where $v' = v + w_j(-p_j + p_{j+1}) + w_j(t + p_{j+1} + s) + q$.

In order to developing an efficient dynamic programming algorithm, the following easy-to-prove lemma is presented first.

Lemma 1: Consider two states (j, t, k, v) and (j', t', k', v') with $0 < j = j' < n$, $k = k'$, $v = v'$, and $t < t'$. As any later schedules generated from (j', t', k', v') cannot be advantage to the corresponding schedules generated from (j, k, t, v), eliminating state

(j',t',k',v') won't lead to any non-optimal solutions to the $1\left|PRM\right|\sum_{S} w_j C_j + \sum_{\bar{S}} e_j + bq$ problem.

The following dynamic programming algorithm starts from an empty state $(0,0,0,0) \in \mathscr{S}^{(0)}$ (where no job has been processed or rejected yet); generates states $(j,t,k,v) \in \mathscr{S}^{(j)}$ by processing (or rejecting) jobs one-by-one; and finds the optimal schedule by selecting the state (n, t, k, v) with the smallest v value over all states in $\mathscr{S}^{(n)}$. Consider the schedule with all jobs rejected, which has the total cost $\sum_J e_j$. The total cost $\sum_J e_j$ can be adopted as the upper bound of the optimal solution value. Thus, the following algorithm will automatically delete any state (j, t, k, v) with $v > \sum_J e_j$, i.e., condition $v' \le \sum_J e_j$ is added into the above three cases.

Algorithm A1

[Initialisation] List all the $\mathscr{S}^{(0)} = \{(0,0,0,0)\}$ and $\mathscr{S}^{(j)} = \varnothing$ for all $j = 1, 2, \cdots, n$.

[StateGeneration] For $j = 0, \cdots, n + 1$, set $\mathscr{T} = \varnothing$ and for each $(j,t,k,v) \in \mathscr{S}^{(j)}$ do:

- If $v' = v + e_{j+1} \le \sum_J e_j$ and reject job $j + 1$, then generate $(j+1,t,k,v')$ and set $\mathscr{T} = \mathscr{T} \cup \{(j+1,t,k,v')\}$.
- If $k < B$ and $v' = v + w_{j+1}(t + p_{j+1}) \le \sum_J e_j$, process job $j + 1$, then generate $(j+1, t+p_{j+1}, k+1, v')$ and set $\mathscr{T} = \mathscr{T} \cup \{(j+1, t+p_{j+1}, k+1, v')\}$. /* Proposition 1.
- If $k = B$, $v' = v + w_{j+1}(t + p_{j+1} + s) + q \le \sum_J e_j$ and $(p_j + \mathrm{s})/w_j \le (p_{j+1} + \mathrm{s})/w_{j+1}$, process job $j + 1$ in the first position after maintenance, then generate $(j+1, t+p_{j+1}+s, 1, v')$ and set $\mathscr{T} = \mathscr{T} \cup \{(j+1, t+p_{j+1}+s, 1, v')\}$ in case that $(p_j + \mathrm{s})/w_j \le (p_{j+1} + \mathrm{s})/w_{j+1}$. /* Proposition 2.
- If $k = B$, $v' = v + w_j(-p_j + p_{j+1}) + w_j(t + p_{j+1} + s) + q \le \sum_J e_j$ and $(p_j + \mathrm{s})/w_j > (p_{j+1} + \mathrm{s})/w_{j+1}$, process job $j + 1$ in the current B position and process job j in the first position after maintenance, then generate $(j+1, t+p_{j+1}+s, 1, v')$ and set $\mathscr{T} = \mathscr{T} \cup \{(j+1, t+p_{j+1}+s, 1, v')\}$ in case that $(p_j + \mathrm{s})/w_j \le (p_{j+1} + \mathrm{s})/w_{j+1}$. /* Proposition 2.

[StateElimination] If $j < n$, for any two states (j, t, k, v) and (j', t', k', v') with $j = j'$, $k = k'$, $v = v'$, and $t < t'$, eliminate state (j', t', k', v') from $\mathscr{T}$, and then set $\mathscr{S}^{(j+1)} = \mathscr{T}$ after conducting all possible eliminations. Otherwise, set $\mathscr{S}^{(n)} = \mathscr{T}$. /* Lemma 1.

[Optimisation] If $\mathscr{S}^{(n)} \neq \varnothing$, find the state (n, t, k, v) with the smallest v value over all states in $\mathscr{S}^{(n)}$. Set the v as the optimal solution value and trace back to obtain the corresponding schedule (S, $\bar{S}$ and the job sequence in S).

Remark 1: Both Proposition 1 and Proposition 2 do not include the position change of jobs from different intervals between maintenance, so that they cannot guarantee the optimal scheduling of all scheduled jobs. Since Algorithm 1 is a dynamic programming algorithm based on property 1 and property 2, it cannot guarantee the optimal scheduling of all the jobs (i.e., scheduled jobs and rejected jobs), but obtain a good feasible solution. The optimal algorithm and the performance bound analysis of Algorithm 1 are still open problems.

3.2 Pseudo-polynomial algorithm for the P2 problem

In this subsection, the Pseudo-polynomial algorithm for the $1\left|PRM, AWP\right|\sum_{S} w_j C_j + \sum_{\bar{S}} e_j + bq$ problem will be considered.

Theorem 2: The $1\left|PRM, AWP\right|\sum_{S} w_j C_j + \sum_{\bar{S}} e_j + bq$ problem is NP-hard.

Proof: Similar with the NP-hard analysis for the $1\left|PRM\right|\sum_{S} w_j C_j + \sum_{\bar{S}} e_j + bq$ problem, the instance I (where job J_i has $p_j = w_j = a_j$, $e_j = Aa_j + a_j^2/2$ and $A = 1/2\sum_{j=1}^{n} a_j$) in Engels et al. (2003) implies that $p_i/w_i = p_j/w_j = 1$ which is clearly satisfies AWP condition ($p_i/w_i \le p_j/w_j$ implies $w_i \ge w_j$). This shows that instance I is also an instance of the $1\left|PRM, AWP\right|\sum_{S} w_j C_j + \sum_{\bar{S}} e_j + bq$ problem. The instance I (where job J_i has $p_j = w_j = a_j$, $e_j = Aa_j + a_j^2/2$ and $A = 1/2\sum_{j=1}^{n} a_j$) can be reduced to the Partitioning problem which is a classic NP hard problem (Engels et al. 2003). Hence, it can be obtained that $1\left|PRM\right|\sum_{S} w_j C_j + \sum_{\bar{S}} e_j + bq$ is also a NP hard problem. □

Let $\bar{C}_j$ be the completion time of job $J_j \in S$ when all the maintenance times are removed from the schedule. As each maintenance activity requires a machine time s, the machine-time cost can be decomposed into three portions such as

$$PC = \sum_{J_j \in S} w_j \overline{C}_j + \sum_{i=1,2,\ldots,b+1} (i-1)s \sum_{J_j \in S_i} w_j + bq. \tag{2}$$

The following two propositions regarding sequencing decision in the processed job set S are easy to prove by the standard interexchange method.

Proposition 3: For jobs in the processed job set S, the optimal sequencing with respect to the first portion $\sum_{J_j \in S} w_j \overline{C}_j$ in (2) is the WSPT order such that jobs with smaller p_j / w_j are processed earlier.

Proposition 4: For jobs in the processed job set S, the optimal sequencing with respect to the second portion $\sum_{i=1,2,\ldots,b+1} (i-1)s \sum_{J_j \in S_i} w_j$ in (2) is the GW (greatest weights) order such that jobs with greater w_j are processed earlier.

Based on Propositions 3 and 4, we can obtain the following proposition.

Proposition 5: For the $1 \left| PRM, AWP \right| \sum_S w_j C_j + \sum_{\overline{S}} e_j + bq$ problem, the jobs in processed job set S can be optimally scheduled in WSPT (or GW) order.

Proof: Based on the agreeable condition of weights and processing times, i.e., for any two jobs J_i and J_j, $p_i / w_i \le p_j / w_j$ implies $w_i \ge w_j$, the WSPT order is equivalent to the GW order. Since b and q denotes the number of maintenance activities and the cost of each maintenance activity, bq is obviously a constant when the set S is given. Based on Propositions 3 and 4, the total scheduled value of jobs in S, i.e., $\sum_{J_j \in S} w_j \overline{C}_j + \sum_{i=1,2,\ldots,b+1} (i-1)s \sum_{J_j \in S_i} w_j + bq$, can be minimised by sequencing jobs in WSPT (or GW) order. □

This proposition indicates that a dynamic programming algorithm, which starts from an initial job sequence with n jobs in their WSPT (or GW) order, can eventually find an optimal schedule by making either processing or rejecting decision on jobs one-by-one. In the following, let the jobs be indexed in their WSPT (or GW) order such as $p_1 / w_1 \le p_2 / w_2 \le \cdots \le p_n / w_n$ (or $w_1 \ge w_2 \ge \cdots \ge w_n$). In the case of $j = n$, (n, t, k, v) denotes a full schedule and also a feasible solution to the $1 \left| PRM, AWP \right| \sum_S w_j C_j + \sum_{\overline{S}} e_j + bq$ problem. For $j < n$, the next unscheduled job J_{j+1} can be scheduled in one of the following three ways.

Case 1 Job J_{j+1} is rejected. State $(j+1, t, k, v')$ is generated, where $v' = v + e_{j+1}$.

Case 2 If $k = B$, job J_{j+1} is processed with a maintenance activity prior to it. State $(j+1, t+s+p_{j+1}, 1, v')$ is generated, where $v' = v + w_{j+1}(p_{j+1} + s + t) + q$.

Case 3 If $k < B$, job J_{j+1} is processed. State $(j+1, t+p_{j+1}, k+1, v')$ is generated, where $v' = v + w_{j+1}(p_{j+1} + t)$.

In order to developing an efficient dynamic programming algorithm, the following easy-to-prove lemma is presented first.

Lemma 3: Consider two states (j, t, k, v) and (j',t',k',v') with $0 < j = j' < n$, $k = k'$, $v = v'$, and $t < t'$. As any later schedules generated from (j',t',k',v') cannot be advantage to the corresponding schedules generated from (j, t, k, v), eliminating state (j',t',k',v') won't lead to any non-optimal solutions to the $1\left|PRM, AWP\right|\sum_{S} w_j C_j + \sum_{\bar{S}} e_j + bq$ problem.

The following dynamic programming algorithm firstly starts from an empty state $(0,0,0,0) \in \mathscr{S}^{(0)}$ where no job has been processed or rejected yet. In the following, we generate states $(j,t,k,v) \in \mathscr{S}^{(j)}$ by processing (or rejecting) jobs one-by-one, and find the optimal schedule by selecting the state (n, t, k, v) with the smallest v value over all states in $\mathscr{S}^{(n)}$. Consider the schedule with all jobs rejected, which has the total cost $\sum_{J} e_j$. In the following proposed algorithm, $\sum_{J} e_j$ is adopted as the upper bound of the optimal solution value. Thus, the following algorithm will automatically delete any state (j, t, k, v) with $v > \sum_{J} e_j$, i.e., condition $v' \le \sum_{J} e_j$ is added into the above three cases.

Algorithm A2

[Initialisation] Set $\mathscr{S}^{(0)} = \{(0,0,0,0)\}$ and $\mathscr{S}^{(j)} = \varnothing$ for all j = 1, 2, …, n.

[StateGeneration] For j = 0, …, n – 1, set $\mathscr{T} = \varnothing$ and for each $(j,t,k,v) \in \mathscr{S}^{(j)}$ do:

- If $v' = v + e_{j+1} \le \sum_{J} e_j$, then generate $(j+1,t,k,v')$ and set $\mathscr{T} = \mathscr{T} \cup \{(j+1,t,k,v')\}$. /* Case 1.
- If $k = B$ and $v' = v + w_{j+1}(t + p_{j+1} + s) + q \le \sum_{J} e_j$, then generate $(j+1, t + p_{j+1} + s, 1, v')$ and set $\mathscr{T} = \mathscr{T} \cup \{(j+1, t + p_{j+1} + s, 1, v')\}$. /* Case 2.
- If $k < B$ and $v' = v + w_{j+1}(t + p_{j+1}) \le \sum_{J} e_j$, then generate $(j+1, t + p_{j+1}, k+1, v')$ and set $\mathscr{T} = \mathscr{T} \cup \{(j+1, t + p_{j+1}, k+1, v')\}$. /* Case 3.

[StateElimination] If $j < n$, for any two states (j, t, k, v) and (j',t',k',v') with $j = j'$, $k = k'$, $v = v'$, and $t < t'$, eliminate state (j',t',k',v') from $\mathscr{T}$, and then set $\mathscr{S}^{(j+1)} = \mathscr{T}$ after conducting all possible eliminations. Otherwise, set $\mathscr{S}^{(n)} = \mathscr{T}$. /* Lemma 3.

[Optimisation] If $\mathscr{S}^{(n)} \neq \varnothing$, find the state ($n$, t, k, v) with the smallest v value over all states in $\mathscr{S}^{(n)}$. Set the $v$as the optimal solution value and trace back to obtain the corresponding schedule (S, $\bar{S}$ and the job sequence in S) as the optimal schedule. If $\mathscr{S}^{(n)} \neq \varnothing$, then the schedule with all rejected jobs is the optimal schedule with the total cost $\sum_J e_j$.

Theorem 3: Algorithm A2 finds an optimal solution to the $1\left|PRM, AWP\right|\sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$ problem in $O\left(n\sum_J e_j\right)$ time. This clarifies that the $1\left|PRM, AWP\right|\sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$ problem is NP-hard in the ordinary sense.

Proof: From Proposition 5, it can be seen that all the processed jobs can be optimally scheduled by in their WSPT (or GW) order. The dynamic programming procedure in Algorithm A2 covers all the decisions of processing or rejecting jobs. It can be seen that the correctness of Algorithm A2 follows directly from the above proposition and discussion.

For the computational complexity of Algorithm A2, there are in total n outer loops. Within each loop, for each state there are at most two options (as Cases 2 and 3 are mutually exclusive). As the [StateElimination] procedure eliminates all unnecessary states, the number of candidate states in $\mathscr{S}^{(j+1)}$ at the beginning of each loop is upper-bounded by $B\sum_J e_j$, where B and $\sum_J e_j$ are the maximum value of k and v, respectively. Therefore, the overall run time of Algorithm A2 is $O\left(2nB\sum_J e_j\right)$, which is indeed $O\left(n\sum_J e_j\right)$. □

3.3 Bounds analysis and FPTAS for the P2 problem

In this subsection, the bound analysis of the $1\left|PRM, AWP\right|\sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$ problem is first proposed, and then an FPTAS will be developed based on the bound analysis.

For the $1\|\sum_S w_j C_j + \sum_{\bar{S}} e_j$ problem, Engels et al. (2003) developed a pseudo-polynomial algorithm and an FPTAS. The design of these time points provides the authors abilities to balance the rum time and solution accuracy. The FPTAS runs in pseudo polynomial time $O\left(n^2/\varepsilon \log \sum_J p_j\right)$. In this section, the static partitioning method (SPM) introduced in Sahni (1976) and the bound improvement procedure (BIP) developed in Chubanov et al. (2006) will be used to convert Algorithm A2 into an FPTAS and this FPTAS runs strongly in polynomial time such as

$O\left(n^2\left(1/\varepsilon+\log\log n\right)\right)$. This improvement is due to the following bounds analysis for the optimal solution value of the $1\left|PRM, ARM\right|\sum_{S} w_jC_j+\sum_{\bar{S}} e_j+bq$ problem. The similar analysis method was firstly developed in Steiner and Zhang (2011). The bound analysis in this paper will adopted this method through integrating SPM and BIP procedures.

Let v^* be the optimal solution value to the $1\left|PRM, AWP\right|\sum_{S} w_jC_j+\sum_{\bar{S}} e_j+bq$ problem. By (1), v^* is the sum of three cost components, i.e., maintenance cost (b^*q), rejection cost $\left(\sum_{\bar{S}^*} e_j\right)$, and machine-time cost $\left(\sum_{S^*} w_jC_j^*\right)$, where * is used to indicate that the calculations are from the same optimal schedule (note that machine-time cost PC in (2) is further divided into maintenance cost and machine-time cost.) As can be seen, $q \le b^*q \le nq$, $e_{[j]} \le \sum_{\bar{S}^*} e_j \le ne_{[j]}$ (where $[j]=\arg\max_{J_j\in\bar{S}^*}\{e_j\}$), and $w_{\{j\}}C^*_{\{j\}} \le \sum_{S^*} w_jC_j^* \le nw_{\{j\}}C^*_{\{j\}}$ (where $\{j\}=\arg\max_{J_j\in S^*}\{w_jC_j^*\}$). This determines a pair of lower and upper bounds, $L^* = q + e_{[j]} + w_{\{j\}}C^*_{\{j\}}$ and $U^* = n(q + e_{[j]} + w_{\{j\}}C^*_{\{j\}})$ such that $L^* \le v^* \le U^* = nL^*$.

As obtaining L^* and U^* is the prior knowledge of S^* and $\bar{S}^*$. But knowing S^* and $\bar{S}^*$ is impossible at this stage. In order to overcome this difficulty, consider a partition J(j) and $\bar{J}([j])$ such that $J([j])\cap\bar{J}([j])=\varnothing$ and $J([j])\cup\bar{J}([j])=J$, where $e_{[j]} \le \max_{J_j\in\bar{J}([j])}\{e_j\}$ and $e_{[j]} > \max_{J_j\in J([j])}\{e_j\}$ (recall $[j]=\arg\max_{J_j\in\bar{S}^*}\{e_j\}$). Thus, $J([j]) \subseteq S^*$ and $\bar{S}^*\subseteq\bar{J}([j])$. Consider sequencing jobs in J([j]) in the WSPT (or GW) order and deploying the same maintenance policy, i.e., reserving a time s after every B jobs. Let $\{\bar{j}\}=\arg\max_{J_j\in J([j])}\{w_jC_j\}$ with respect to the same schedule. $J([j]) \subseteq S^*$ gives $w_{\{\tilde{j}\}}C_{\{\tilde{j}\}} \le w_{\{j\}}C^*_{\{j\}}$, and therefore proves $w_{\{\tilde{j}\}}C_{\{\tilde{j}\}}+e_{[j]} \le \sum_{S^*} w_jC_j^*+\sum_{\bar{S}^*} e_j^*$. Next, $\sum_{S^*} w_jC_j^*+\sum_{\bar{S}^*} e_j^* \le nw_{\{\tilde{j}\}}C_{\{\tilde{j}\}}+ne_{[j]}$ is proved in two different cases.

Case 1 assumes $\sum_{S^*} w_jC_j^* \le nw_{\{\bar{j}\}}C_{\{\bar{j}\}}$. As $\sum_{\bar{S}^*} e_j^* \le ne_{[j]}$, the proof is obvious. In Case 2, $\sum_{S^*} w_jC_j^* > nw_{\{\tilde{j}\}}C_{\{\tilde{j}\}}$ is assumed. This can occur only when $S^*-J([j])\ne\varnothing$.. As v^* is the optimal solution value, for any job $j \in S^* - J([j])$, it is sure that $e_j \ge w_jC_j^*$. The definition of $J([j])$ and $\bar{J}([j])$ gives $e_j < e\{[j]\}$. This completes the proof for Case 2 as follows:

$$\sum_{S^*} w_jC_j^*+\sum_{\bar{S}^*} e_j^* \le \sum_{J([j])} w_jC_j^*+\sum_{\bar{S}^*+(S^*-J([j]))} e_j^* \tag{3}$$

$$\leq \sum_{J([j])} w_j C_j + \sum_{\bar{S}^* + (S^* - J([j]))} e_j^* \tag{4}$$

$$\leq n w_{\{\tilde{j}\}} C_{\{\tilde{j}\}} + n e_{[j]}. \tag{5}$$

Taking into account the maintenance cost, a pair of bounds $L([j]) = q + e_{[j]} + w_{\{j\}}C_{\{j\}}$ and $U([j]) = n(q + e_{[j]} + w_{\{j\}}C_{\{j\}})$ with $L([j]) \leq v^* \leq U([j]) = nL([j])$ are determined, where the only prior knowledge is $[j]$. Next, it is shown that the bounds of v^* can be determined without knowing $[j]$.

Suppose jobs in J have K number of distinct, positive job rejection costs, $e_{[1]} < e_{[2]} < \cdots < e_{[K]}$. In particular, $e_{[0]} = 0$ is used to create the partition $J([0]) = J$ and $\bar{J}([0]) = \varnothing$, which is used to describe the schedule without rejected jobs. Obviously, $e_{[j]}$ takes one of these $k + 1$ distinct rejection costs. Note that each of $e_{[i]}$, $i = 0, 1, \cdots, K$ specifies a type of feasible schedules for the $1\left|PRM, AWP\right|\sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$ problem, where $e_{[i]}$ is the largest rejection cost over all rejected jobs. Let $v_{[i]}$ be the smallest solution value over all feasible schedules specified by $e_{[i]}$. A pair of bounds can be easily determined such as $L(i) \leq v_{[i]} \leq U(i) = nL(i)$, $L([i]) = q + e_{[i]} + w_{\{j\}}C_{\{j\}}$ and $\left\{\bar{i}\right\} = \arg\max_{J_i \in J([i])} \{w_i C_i\}$. For the calculation of $v^* = \min_{i=0,1,\cdots,K} \{v_{[i]}\}$, it takes n calculations for each $\left\{\bar{i}\right\}$. Based on the above discussion, the following lemma can be obtained.

Lemma 4: When $J([i])$ are processed in the WSPT (or GW) order with a piece-rate maintenance activity inserted after every B jobs, a pair of bounds $L \leq v^* \leq U$ with $U = nL$ can be determined in $O(n^2)$ time by $L = \min_{i=0,1,\cdots,K} \{L([i])\}$, where $L([i]) = q + e_{[i]} + w_{\{j\}}C_{\{j\}}$, $e_{[i]} \leq \max_{J_i \in \bar{J}([i])} \{e_i\}$ and $e_{[i]} > \max_{J_i \in J([i])} \{e_i\}$, and $\left\{\bar{i}\right\} = \arg\max_{J_i \in J([i])} \{w_i C_i\}$.

By directly implementing SPM in Sahni (1976) on Algorithm A2, the above bounds ($L \leq v^* \leq U = nL$) will lead to an approximation algorithm, which for any given $\varepsilon > 0$ will find an $(1 + \varepsilon)$-approximation solution to the $1\left|PRM, ARM\right|\sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$ problem in $O(n^3/\varepsilon)$ time. In order to clearly describe the calculation procedure, the following approximation algorithm (denoted by $A(Z, n, \varepsilon)$, where Z is the upper bound, Z/n is the lower bound, and $\varepsilon \times Z/n$ is maximum cumulative error introduced into the found solution) is presented as follows.

Algorithm A(X, n, ε)

[Initialisation] Set $\mathscr{S}^{(0)} = \{(0,0,0,0)\}$ and $\mathscr{S}^{(j)} = \varnothing$ for all $j = 1, 2, \ldots, n$.

[Partitioning] Partition $[0, Z]$ into $\lceil n/\varepsilon \times n \rceil$ equal subintervals of size $\varepsilon/n \times Z/3$ with the last one possibly smaller (the latter n in $\lceil n/\varepsilon \times n \rceil$ is due to Z/n is the lower bound).

[StateGeneration] Except for replacing the condition $v' \le \sum_J e_j$ with $v' \le Z$, do the same as in Algorithm A.

[StateElimination] If $j < n$, for the states (j, t, k, v) with the same k value and $v \le Z$ values falling in the same subinterval, keep only the one with the smallest t value, and set $\mathscr{S}^{(j+1)} = \mathscr{T}$ after all of these operations. Otherwise, set $\mathscr{S}^{(n)} = \mathscr{T}$.

[StateSelection] If $\mathscr{S}^{(n)} \ne \varnothing$, find the state ($n$, t, k, v) with the smallest v value. Set v as the solution value and trace back to obtain the corresponding schedule (S, $\overline{S}$ and the job sequence in S).

Remark 2: In [Partitioning], the interval [0, Z] is divided into $\lceil n^2/\varepsilon \rceil$ subintervals as the length of each subinterval has to be no more than $\varepsilon Z/n^2$, which is the maximum error allowed to be introduced in each iteration. This indicates that the total cumulative error of n iterations will be no more than $\varepsilon Z/n^2$, which achieves the designated approximation ratio ε. The size of each subinterval is the control factor of solution accuracy and the number of subintervals is the control factor of algorithm efficiency. These two factors, however, are linked together via the structure of the bounds. This shows that the quality of the bounds (i.e., how tight of the bounds) is the key to the algorithm efficiency and solution accuracy.

In Chubanov et al. (2006), a binary searching procedure BIP is introduced to make improvement on objective bound, i.e., improving the initial bound [L, $U = nL$] to a tighter bound $[L', U' = 3L']$. Using $[L', U' = 3L']$, Algorithm $A(Z = U', n = 3, \varepsilon)$ partitions $[0, U']$ into $\lceil n/\varepsilon \times 3 \rceil$ equal subintervals of size $\varepsilon/n \times U'/3$ with the last one possibly smaller. Thus, the achieved approximation ratio is the same ε, but the run time is improved to $O(n/\varepsilon \times 3n) = O(n^2/\varepsilon)$ with an order of n reduction. In order to obtain $[L', U' = 3L']$, BIP calls a slightly modified version of Algorithm A(Z, n, ε), denoted by $A'(X, 2, 2)$, repeatedly to make guesses on the bounds of v^*. The modification occurs in the last procedure [StateSelection]. Instead of tracing back the found solution value v to obtain the corresponding schedule, Algorithm $A'(X, 2, 2)$ report $v^* > 2X/3$ if $v > X$, or report $v^* \le X$ otherwise. For detailed proof of this modification, interested readers are referred to Chubanov et al. (2006). It is clear that Algorithm $A'(X, 2, 2)$ runs in $O(n/2 \times 2n) = O(n^2)$ time, as $\varepsilon = 2$ and the guessed upper bound X is two folds of the lower bound. In the following, Algorithm A_ε is proposed through combining the initial bounds determination, the BIP implementation, and the final approximation algorithm.

Algorithm A_ε

[BoundsCalculation] Determine initial bounds [L, $U = nL$] with $L = \min_{i=0,1,\cdots,K} \{L([i])\}$.

[BIPImplementation] Conduct binary search on [0, $\log_2 n$] (Chubanov et al. 2006). Note that the original bounds can be re-written by $L = 2^0 \times L$ and $U = nL = 2^{\log_2 n} \times L$.

Set $L' = 2^{\lceil \log_2 n \rceil} L/3$, $l_1 = 0$, and $l_2 = [\log_2 n]$.

Set $k = \lceil (l_1 + l_2)/2 \rceil$ and $X = 2^{k-1} L'$.

Run Algorithm $A'(X, 2, 2)$ on the $1\big|PRM, ARM\big|\sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$ problem.

For $v > X$ ($v^* > 2X/3$), , if $l_2 = k$, then stop; otherwise set $l_1 = k$ and go to 2.

For $v \le X$ ($v^* \le X$), if $l_2 = k$, then set $L' = X/3$ and stop; otherwise set $l_2 = k$, $L' = X/3$ and go to 2.

[Approximation] Run Algorithm $A(U', 3, \varepsilon)$ on the $1\big|PRM, ARM\big|\sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$ problem.

Theorem 4: For any given $\varepsilon > 0$, Algorithm A_ε finds an $(1 + \varepsilon)$-approximation solution value such as $v \le (1 + \varepsilon)v^*$ for the $1\big|PRM, AWP\big|\sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$ problem. The run time is $O(n^2 \max\{\log\log n, 1/\varepsilon\})$ time. This indicates that Algorithm A_ε is an FPTAS for the $1\big|PRM, AWP\big|\sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$ problem.

Proof: The correctness of Algorithm A_ε directly follows the above discussions and the proofs in Steiner and Zhang (2012). The procedure [BoundsCalculation] runs in $O(n^2)$ time. In [BIPImplementation], Algorithm $A(X, 2, 2)$ used no more than loglog$n$ time, and Algorithm $A(X, 2, 2)$ runs in $O(n^2)$ time. In [Approximation] step, $A(U', 3, \varepsilon)$ runs in $O(n^2/\varepsilon)$ time. Thus, the overall run time is $O(n^2[1 + \log_2 \log_2 n + 1/\varepsilon]) = O(n^2 \max\{\log\log n, 1/\varepsilon\})$. □

4 Optimal solutions for the P3-P4 problems

In this section, two special cases of, one with equal weights $\left(1 \mid PRM, w_j = w \mid w\sum_S C_j + \sum_{\bar{S}} e_j + bq\right)$ and one with equal processing times $\left(1 \mid PRM, p_j = p \mid \sum_S w_j C_j + \sum_{\bar{S}} e_j + bq\right)$, are investigated.

4.1 Optimal solution for the P3 problem

Given a schedule, the total cost can be described as $TC = PC + RC$, where $RC = \sum_{J_j \in \bar{S}} e_j$,

$$PC = w \sum_{J_j \in S} \bar{C}_j + \sum_{i=1,2,\cdots,b+1} (i-1)s \sum_{J_j \in S_i} w + bq, \tag{6}$$

where PC modifies the machine-time cost defined in (2) to reflect the fact that all the jobs have equal weights. Without considering the maintenance-related cost $\sum_{i=1,2,\cdots,b+1}(i-1)s\sum_{J_j\in S_i} w$ and bq, (6) shows that a rejected job $J_j \in \bar{S}$ contributes e_j to the total cost and a processed job $J_j \in S$ contributes $w\left(|S|-[j]+1\right)p_j$, where $[j]$ indicates that job J_j is the $[j]$th processed job. Let MC be the sum of the two maintenance-related costs above, i.e., the sum of the last two items in (6). It can be re-written as

$$MC = Bws\sum_{i=1}^{\lceil |S|/B-1\rceil}(i-1) + ws(|S|-\lceil |S|/B\rceil B)(\lceil |S|/B\rceil -1) + (\lceil |S|/B\rceil -1)q, \tag{7}$$

which shows that MC depends entirely on $|S|$ (the total number of processed jobs).

Given a schedule, let r be the number of rejected jobs (i.e., $|\bar{S}|=r$), which are assumed to occupy the $n-r+1$ to n positions. The first $n-r$ positions are reserved for the rest $n-r$ processed jobs (i.e., $|S|=n-r$). Therefore, the total cost can be written by

$$TC(r,S,\bar{S}) = RPC(r,S,\bar{S}) + MC(r), \tag{8}$$

where $S = \{J_j \mid [j] = 1, 2, \cdots, n-r\}$,

$$\bar{S} = \{J_j \mid [j] = n-r+1, n-r+2, \cdots, n\}.$$

$$RPC(r,S,\bar{S}) = w\sum_{[j]=1}^{n-r}(n-r-[j]+1)p_j + \sum_{[j]=n-r+1}^{n} e_j \tag{9}$$

$$\begin{aligned} MC(r) = {} & \left(\lceil (n-r)/B\rceil -1\right)q + Bws\sum_{i=1}^{\lceil (n-r)/B\rceil -1}(i-1) \\ & + ws\left(n-r-\lceil (n-r)/B\rceil B\right)(\lceil (n-r)/B\rceil -1) \end{aligned} \tag{10}$$

Let $RPC^*(r)$ be the smallest $RPC(r,S,\bar{S})$ cost over all such partitions $\bar{S}$ and S, and all possible orders in S gives the smallest cost of the $1 \mid PRM, w_j = w \mid w\sum_S C_j + \sum_{\bar{S}} e_j + bq$ problem.

$$TC^* = \min_{r=0,1\cdots,n}\left\{RPC^*(r) + MC(r)\right\} \tag{11}$$

$$= \min_{r=0,1\cdots,n}\left\{MC(r) + \min_{S=\{[1],\cdots,[n-r]\},\bar{S}=\{[n-r+1],\cdots,[n]\}}\left\{RPC(r,S,\bar{S})\right\}\right\} \tag{12}$$

Considering $r \in \{1, 2, \cdots, n-1\}$, finding $RPC^*(r)$ and the associated $S^*(r)$ and $\bar{S}^*(r)$ is an assignment problem, which can be formulated as a mixed-integer linear programming:

$$\text{MIL(r)} \quad \min \quad \sum_{j=1}^{n} x_{jk}c_{jk}^{r} \tag{13}$$

$$\text{st:}\sum_{k=1}^{n} x_{jk} \leq 1, \qquad \forall j \tag{14}$$

$$\sum_{j=1}^{n} x_{jk} \leq 1, \qquad \forall k \tag{15}$$

$$x_{ij} \in \{0,1\}, \qquad \forall j,k, \tag{16}$$

where the decision variables are x_{jk} = 1 if job J_j is scheduled in the kth position and x_{jk} = 0 otherwise. The cost c_{jk}^{r} can be defined by $c_{jk}^{r} = e_j$ if $k > n - r$ (i.e., J_k is rejected) and $c_{jk}^{r} = w\sum_{k=1}^{n-r}(n-r-k+1)p_j$ if $k \leq n-r$ (i.e., J_k is processed). It is not hard to see that the constraint matrix of MIL(r) holds the totally-modularity property. Therefore, the linear relaxation of MIL(r) can be solved in polynomial time with integer solution values (x_{jk}^{*}) automatically returned. This gives the optimal solution $RPC^{*}(r) = \sum_{j=1}^{n} x_{jk}^{*} c_{jk}^{r}$, $S^{*}(r) = \{J_j \mid x_{jk}^{*} = 1 \; and \; k \leq n-r\}$, and $\bar{S}^{*}(r) = \{J_j \mid x_{jk}^{*} = 1 \; and \; k > n-r\}$.

Proposition 5 shows that sequencing processed jobs in the WSPT (or GW) order is optimal. For this special case with all equal weights, these two orders are reduced into the SPT (shortest processing time first) order. Based on this, let jobs be indexed such as $p_1 \leq p_2 \leq \cdots \leq p_n$. Now, consider two extreme partitions: $r = 0$ and $r = n$. For $r = 0$ (i.e., $S = J$), the smallest total cost can be written by

$$\begin{aligned} TC^{0} &= w\sum_{j=1}^{n}(n-j+1)p_j + \left(\lceil n/B \rceil - 1\right)q \\ &\quad - Bws\sum_{i=1}^{\lceil n/B \rceil - 1}(i-1) - ws\left(n - \lceil n/B \rceil B\right)(\lceil n/B \rceil - 1) \end{aligned} \tag{17}$$

For r = n (i.e., $S = \varnothing$), the smallest total cost can be calculated by $TC^{n} = \sum_{j=1}^{n} e_j$. Next, the algorithm (denoted by A_w) for the $1 \mid PRM, w_j = w \mid w\sum_{S} C_j + \sum_{\bar{S}} e_j + bq$ problem is presented.

Algorithm A_w

[Initialisation] Set $TC = \min\{TC^0, TC^n\}$ and $r = 1$.

[r-rejectedJobs] Calculate c_{jk}^{r}, formulate and solve MIL(r) to obtain $RPC^*(r)$.

- If $RPC^*(r) + MC(r) < TC$, then set $TC = RPC^*(r) + MC(r)$;

- If $r < n - 1$, then set $r = r + 1$ and go to [r-rejectedJobs].

[Solution] Set $TC^* = TC$ and trace back to obtain S^*, $\bar{S}^*$, and job sequences in S^*.

Theorem 5: Algorithm A_w is a polynomial algorithm such that it finds an optimal solution to the $1 \mid PRM, w_j = w \mid w\sum_S C_j + \sum_{\bar{S}} e_j + bq$ problem in $O(n^4)$ time.

Proof: By Algorithm A_w, the $1 \mid PRM, w_j = w \mid w\sum_S C_j + \sum_{\bar{S}} e_j + bq$ problem is essentially converted into the n iterations of the assignment problem MIL(r). Since the assignment problem can be optimally solved (Kuhn, 1955), it is clear that the $1 \mid PRM, w_j = w \mid w\sum_S C_j + \sum_{\bar{S}} e_j + bq$ problem can be optimally solved by Algorithm A_w. Regarding run time, TC is initialised to the smaller objective value of the schedule with zero-rejected job (TC^0) and the schedule with all-rejected jobs (TC^n). This calculation takes $O(n)$ time. For each r, it takes $O(n^2)$ time to calculate c_{jk}^r, and it takes $O(n^3)$ time to formulate and solve the assignment problem MIL(r) because that Kuhn (1955) showed that the computation complexity of solving an $n \times n$ assignment problem is $O(n^3)$. [r-rejectedJobs] goes through $n - 1$ different r values. The overall run time of this step is $O(n^4)$. Given that [Solution] runs in $O(n^3)$ time, it is proved that Algorithm A_w runs in $O(n^4)$ time. □

4.2 Optimal solution for the P4 problem

The methods (MIL(r) and Algorithm A_w) above can be modified and applied on the equal-processing-time case $\left(1 \mid PRM, p_j = p \mid \sum_S w_j C_j + \sum_{\bar{S}} e_j + bq\right)$. The run time would be the same $O(n^4)$. In what follows, however, a modified version of Algorithm A2 is developed to solve the latter equal-processing-time case, and the run time can be reduced to $O(n^2)$.

Similarly to Lemma 2, it can be shown that for any two states (j, t, k, v) and (j', t', k', v') with $j = j'$, $t = t'$, $k = k'$, and $v < v'$, any later schedules generated from (j', t', k', v') cannot be advantage to the corresponding schedules generated from (j, t, k, v). Based on this, Algorithm A2 can be modified such that [StateElimination] is replaced with the followings: If $j < n$, for any two states (j, t, k, v) and (j', t', k', v') with $j = j'$, $t = t'$, $k = k'$, and $v < v'$, eliminate state (j', t', k', v') from $\mathscr{T}$, and set $\mathscr{S}^{(j+1)} = \mathscr{T}$ after all possible eliminations. Otherwise, set $\mathscr{S}^{(n)} = \mathscr{T}$ directly. The modified version is denoted by Algorithm A_p.

As this modification changes the state space from value-based to time-based, the number of states in each $\mathscr{S}^{(j)}$ is given by the number of possible job completion times (denoted by π). Consider state (j, k, t, v) (where $0 < j \le n$ and $0 \le k \le B$), possible t values are:

$t = kp$ (if $j-k$ jobs are rejected),

$t = (B+k)p+s$ (if $j-k-B$ jobs are rejected),

$t = (2B+k)p+2s$ (if $j-k-2B$ jobs are rejected),

$\cdots,$

$t = (B\lfloor j/B \rfloor + k)p + s\lfloor j/B \rfloor$ (if $j-k-B\lfloor j/B \rfloor$ jobs are rejected).

Thus, $\pi(j,k) = \lfloor j/B \rfloor + 1$. As k takes B different values for each j, in $\mathscr{S}^{(j)}$ the number of states is upper-bounded by $B(\lfloor j/B \rfloor + 1) < n + B$. Considering j takes $n + 1$ values, the following corollary is straightforward.

Theorem 6: The $1 \mid PRM, p_j = p \mid \sum_S w_j C_j + \sum_{\bar{S}} e_j + bq$ problem can be optimally solved in $O(n^2)$ time by Algorithm A_p.

Proof: Algorithm A_p is the modified version of Algorithm A2 for the special case with $p_j = p$. The correctness of Algorithm A_p follows directly from the correctness proof of Algorithm A2. Similar with the analysis for the computational complexity of Algorithm A2, it can be obtained that the computational complexity of Algorithm A_p is $O\left(n\sum_J p_j\right)$. Since all the jobs have the same processing times, i.e., $p_j = p$, we have $\sum_J p_j = np$. The computational complexity $O\left(n\sum_J p_j\right)$ can be reduced to $O(n^2)$because that $\sum_J p_j = np$ and p is a constant. Therefore, the computational complexity of Algorithm A_p is $O(n^2 p)$. □

5 Conclusions

In this paper, we consider the single-machine scheduling problems with rejection and piece-rate maintenance. The considered objective is to minimise the sum of weighted completion times, rejection cost, maintenance cost. We address four different problems for treating four different cases. For the first problem, it is proved to be NP-hard, and its approximated solving algorithm is developed. For the second problem with agreeable weights and processing times, after addressing its NP-hardness, a pseudo-polynomial time algorithm is presented to establish that the problem is NP-hard only in the ordinary sense, and then the algorithm is further converted into an FPTAS. In the last, the other two special cases, one with all equal weights and one with all equal processing times, are addressed to be optimally solvable in polynomial time.

For future research, it would be interesting to find out whether the first problem is NP-hard in the ordinary sense or in the strong sense and to include the analysis of scheduling problems with other machine environments (e.g. parallel machine, flowshop and jobshop). In addition, the efficient approximation algorithm and its performance bound analysis for the first problem will be worth of further study by scholars.

Acknowledgements

This work is supported by the National Nature Science Foundation of China (71774080, 71501098 and 71834003), Postdoctoral Science Foundation Funded Project (2016M590453, 1601245C, 2018T110501), Science Foundation of Ministry of Education of China (17YJC630205). In memoriam: Tragically, Dr. Rui Zhang, who contributed to this study, went away due to the cancer disease before this manuscript could be published. We wish to honour his memory.

References

Bartal, Y., Leonardi, S., Marchetti-Spaccamela, A., Sgall, J. and Stougie, L. (2000). 'Multiprocessor scheduling with rejection', *SIAM Journal on Discrete Mathematics*, Vol. 13, No. 1, pp.64–78.

Cheng, T.C.E. and Wang, G. (1999) 'Two-machine flowshop scheduling with consecutive availability constraints', *Information Processing Letters*, Vol. 71, No. 2, pp.49–54.

Chubanov, S., Kovalyov, M.Y. and Pesch, E. (2006) 'An fptas for a single-item capacitated economic lot-sizing problem with monotone cost structure', *Mathematical programming*, Vol. 106, No. 3, pp.453–466.

Cui, W.W. and Lu, Z. (2017) 'Minimizing the makespan on a single machine with flexible maintenances and jobs' release dates', *Computers & Operations Research*, Vol. 80, No. 1, pp.11–22.

Engels, D.W., Karger, D.R., Kolliopoulos, S.G., Sengupta, S., Uma, R. and Wein, J. (2003) 'Techniques for scheduling with rejection', *Journal of Algorithms*, Vol.49, No. 1, pp.175–191.

Gawiejnowicz, S. (2007) 'Scheduling deteriorating jobs subject to job or machine availability constraints', *European Journal of Operational Research*, Vol. 180, No. 1, pp.472–478.

Graham, R.L., Lawler, E.L., Lenstra, J.K. and Rinnooy Kan, A.H.G. (1979) 'Optimization and approximation in deterministic machine scheduling: a survey', *Annals of Discrete Mathematics*, Vol. 5, No. 1, pp.287–326.

Guo, L., Liu, M., Sethi, S.P. and Xu, D. (2017) 'Parallel-machine scheduling with machine-dependent maintenance periodic recycles', *International Journal of Production Economics*, Vol. 186, No. 10, pp.1–7.

Hadidi, L.A., Al-Turki, U.M. and Rahim, M.A. (2011) 'An integrated cost model for production scheduling and perfect maintenance', *International Journal of Mathematics in Operational Research*, Vol. 3, No. 4, pp.395–413.

Kannan, G., Saravanakumar, G. and Saraswathi, M. (2018) 'Two-degree of freedom PID controller in time delay system using hybrid controller model', *International Journal of Automation and Contro*l, Vol. 12, No. 3, pp.399–426.

Kong, M., Liu, X., Pei, J., Zhou, Z. and Pardalos, P.M. (2019) 'Parallel-batching scheduling of deteriorating jobs with non-identical sizes and rejection on a single machine', *Optimization Letters*, Vol. 3, No. 1, pp.1–15.

Kuhn, H.W. (1955) 'The hungarian method for the assignment problem', *Naval Research Logistics Quarterly*, Vol. 2, Nos. 1–2, pp.83–97.

Kumar, S. and Lad, B.K. (2017) 'Integrated production and maintenance planning for parallel machine system considering cost of rejection', *Journal of the Operational Research Society*, Vol. 68, No. 7, pp.834–846.

Li, C., Wang, C. and Luo, Y. (2020) 'An efficient scheduling optimization strategy for improving consistency maintenance in edge cloud environment', *The Journal of Supercomputing*, DOI: 10.1007/s11227-019-03133-9.

Liu, Q., Ming, D. and Chen, F.F. (2018) 'Single-machine-based joint optimization of predictive maintenance planning and production scheduling', *Robotics and Computer-Integrated Manufacturing*, Vol. 51, No. 1, pp.238–247.

Lu, L., Cheng, T.C.E., Yuan, J. and Zhang L. (2009) 'Bounded single-machine parallel-batch scheduling with release dates and rejection', *Computers and Operations Research*, Vol. 36, No. 10, pp.2748–2751.

Mohamed, A., Ren, J., El-Gindy, M., Lang, H. and Ouda, A.N., (2018a) 'Literature survey for autonomous vehicles: sensor fusion, computer vision, system identification and fault tolerance', *International Journal of Automation and Control*, Vol. 12, No. 4, pp.555–581.

Mohamed, A., Ren, J., Lang, H. and El-Gindy, M. (2018b) 'Optimal path planning for an autonomous articulated vehicle with two trailers', *International Journal of Automation and Control*, Vol. 12, No. 3, pp.449–465.

Omar, R. and Shaik, M. (2019) 'Integrated scheduling of production and maintenance for continuous plants: conditional sequencing and approximate modeling of storage',*Industrial & Engineering Chemistry Research*, Vol. 58, No. 41, pp.19100–19121.

Ou, J., Zhong, X. and Wang, G. (2015) 'An improved heuristic for parallel machine scheduling with rejection', *European Journal of Operational Research*, Vol. 241, No. 3, pp.653–661.

Pati, A. and Negi, R., (2019) 'Super-twisting algorithm-based integral sliding mode control with composite nonlinear feedback control for magnetic levitation system', *International Journal of Automation and Control*, Vol. 13, No. 6, pp.717–734.

Qi, X. (2011) 'Outsourcing and production scheduling for a two-stage flow shop', *International journal of production economics*, Vol. 129, No. 1, pp.43–50.

Sahni, S.K. (1976) 'Algorithms for scheduling independent tasks', *Journal of the ACM (JACM)*, Vol. 23, No. 1, pp.116–127.

Selvarajah, E. and Zhang, R. (2014) 'Supply chain scheduling to minimize holding costs with outsourcing', *Annals of Operations Research*, Vol. 217, No. 1, pp.479–490.

Shabtay, D. (2014) 'The single machine serial batch scheduling problem with rejection to minimize total completion time and total rejection cost', *European Journal of Operational Research*, Vol. 233, No. 1, pp.64–74.

Shabtay, D., Gaspar, N. and Kaspi, M. (2013) 'A survey on offline scheduling with rejection', *Journal of Scheduling*, Vol. 16, No. 1, pp.3–28.

Somasundaram, S. and Benjanarasuth, T. (2019) 'CDM-based two degree of freedom PI controller tuning rules for stable and unstable FOPTD processes and pure integrating processes with time delay', *International Journal of Automation and Control*, Vol. 13, No. 3, pp.263–281.

Steiner, G. and Zhang, R. (2011) 'Minimizing the weighted number of tardy jobs with due date assignment and capacity-constrained deliveries', *Annals of Operations Research*, Vol. 191, No. 1, pp.171–181.

Thevenin, S., Zufferey, N. and Widmer, M. (2015) 'Metaheuristics for a scheduling problem with rejection and tardiness penalties', *Journal of Scheduling*, Vol. 18, No. 1, pp.89–105.

Wang, D., Yin, Y. and Liu M. (2016) 'Bicriteria scheduling problems involving job rejection, controllable processing times and ratemodifying activity', *International Journal of Production Research*, Vol. 54, No. 12, pp.3691–3705

Wang, D., Yin, Y. and Cheng, T.C.E. (2018) 'Parallel-machine rescheduling with job unavailability and rejection', *Omega*, Vol. 81, No. 1, pp.246–260.

Xiong, X., Wang, D., Cheng, T.C.E., Wu, C.C. and Yin, Y. (2018) 'Single-machine scheduling and common due date assignment with potential machine disruption', *International Journal of Production Research*, Vol. 56, Nos. 3–4, pp.1345–1360.

Xu, D., Sun, K. and Li, H. (2008) 'Parallel machine scheduling with almost periodic maintenance and non-preemptive jobs to minimize makespan', *Computers & Operations Research*, Vol. 35, No. 4, pp.1344–1349.

Xu, D., Wan, L., Liu, A. and Yang, D. L. (2015) 'Single machine total completion time scheduling problem with workload-dependent maintenance duration', *Omega*, Vol. 52, No. 1, pp.101–106.

Xu, D., Yin, Y. and Li, H. (2009) 'A note on scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan', *European Journal of Operational Research*, Vol. 197, No. 2, pp.825–827.

Xu, Z. and Xu D. (2018) 'Single-machine scheduling with workload-dependent tool change durations and equal processing time jobs to minimize total completion time', *Journal of Scheduling,* Vol. 21, No. 4, pp.461–482.

Xue, P., Zhang, Y. and Yu, X. (2014) 'Single-machine scheduling with piece-rate maintenance and interval constrained position-dependent processing times', *Applied Mathematics and Computation*, Vol. 226, pp.415–422.

Yin, Y., Cheng, T. C. E. and Wang, D. (2015) 'Improved algorithms for single-machine serial-batch scheduling with rejection to minimize total completion time and total rejection cost', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 46, No. 11, pp.1578–1588.

Yin, Y., Cheng, T.C.E. and Wang, D.J. (2016) 'Rescheduling on identical parallel machines with machine disruptions to minimize total completion time', *European Journal of Operational Research*, Vol. 252, No. 3, pp.737–749.

Yin, Y., Wang, Y., Cheng, T.C.E., Liu, W. and Li, J. (2017) 'Parallel-machine scheduling of deteriorating jobs with potential machine disruptions', *Omega*, Vol. 69, No. 1, pp.17–28.

Yu, X., Zhang, Y., Xu, D. and Yin, Y. (2013) 'Single machine scheduling problem with two synergetic agents and piece-rate maintenance', *Applied Mathematical Modelling*, Vol. 37, No. 3, pp.1390–1399.

Zhang, L., Lu, L. and Yuan J. (2010) 'Single-machine Scheduling under the Job Rejection Constraint', *Theoretical Computer Science*, Vol. 411, Nos. 16–18, pp.1877–1882.

Zhang, M. and Luo, C. (2013) 'Parallel-machine scheduling with deteriorating jobs, rejection and a fixed non-availability interval', *Applied Mathematics and Computation*, Vol. 224, No. 3, pp.405–411.

Zhong, X., Ou, J. and Wang, G. (2014) 'Order acceptance and scheduling with machine availability constraints', *European journal of operational research*, Vol. 232, No. 3, pp.435–441.