J. Appl. Math. Comput. (2015) 48:585–605
DOI 10.1007/s12190-014-0820-3

ORIGINAL RESEARCH

Deteriorating jobs scheduling on a single machine with release dates, rejection and a fixed non-availability interval

Wei-Xuan Li · Chuan-Li Zhao

Received: 28 March 2014 / Published online: 26 August 2014

Abstract This paper studies single machine scheduling with a fixed non-availability interval. The processing time of a job is a linear increasing function of its starting time, and each job has a release date. A job is either rejected by paying a penalty cost or accepted and processed on the machine. The objective is to minimize the makespan of the accepted jobs and the total rejection penalties of the rejected jobs. We present a fully polynomial-time approximation scheme for the problem. We also show that the special case without non-availability interval can be solved using the same method with a lower order.

Keywords Scheduling · Single machine · Deteriorating jobs · Release dates · Rejection · Non-availability interval

Mathematics Subject Classification 90B35

1 Introduction

Scheduling problems have been extensively studied in various machine environments and performance measures. In most classical scheduling problems, the machine is available all the time. Actually, however, a machine may be unavailable in some parts of the scheduling period due to preventive maintenance or tool change and so on. Due to its strong background in industrial applications, machine scheduling with

W.-X. Li (✉) · C.-L. Zhao
School of Mathematics and System Science, Shenyang Normal University,
No. 253 Huanghe North Street, Huanggu District, Shenyang 110034, China
e-mail: liweixuancz@126.com

C.-L. Zhao
e-mail: zhaochuanli@synu.edu.cn

availability constraints has received increasing attention. Two cases, resumable and non-resumable, were defined in the literature (Lee [9]). A job is called resumable if it cannot finish before the unavailable period of a machine and can continue after the machine is available again. On the other hand, a job is called non-resumable if it has to restart, rather than continue. Li and Fan [11] studied the non-resumable version of the scheduling problem on a single machine subject to availability constraints. The objective is to minimize the total weighted completion time. They proposed a pseudo-polynomial-time algorithm and a FPTAS for the problem with a single non-availability interval. Ma et al. [18] provided a survey of different problems with machine availability constraints. Zhao et al. [29] considered two parallel machines scheduling problem where one machine is not available in a fixed time period, they gave a FPTAS for the problem. Shen et al. [21] and Wang et al. [22] focused on parallel-machine scheduling with non-simultaneous machine available time. Zhao and Tang [28] considered a parallel-machine scheduling problem. Each machine is not available in a specified time period. They presented a pseudo-polynomial dynamic programming algorithm.

Furthermore, recent empirical studies have verified that the job processing times are increasing functions of their starting times, such as cleaning tasks, scheduling maintenance and so on, where any delay in starting to process a job increases the job's processing time. The phenomenon is "deteriorating jobs". Hsu et al. [5] analyzed linear deteriorating jobs scheduling with due-date assignment and maintenance activity on a single machine. The objective is to minimize the total of earliness, tardiness and due-date cost. They showed that the problem can be solved in polynomial time. Wang et al. [22] studied single-machine scheduling problems with a time-dependent deterioration, where the job processing time is defined by an increasing function of the total normal processing time of jobs in front of it in a sequence. They constructed a mixed integer programming formulation for the problem. Wang and Wang [23] considered scheduling problem with convex resource dependent processing times and deteriorating jobs, in which the job processing time is a function of its starting time and its convex resource allocation. Liu et al. [15] stressed a single-machine common due-window assignment scheduling problem with deteriorating jobs. The job processing times are functions of their starting times and job-dependent deterioration rates that are related to jobs and are not all equal. Cheng et al. [1] made a concise survey of scheduling with time-dependent processing time.

Additionally, in many cases jobs rejection may be considered. For example, to obtain the maximum profits, the manufacturer often rejects some jobs which have the larger processing times and bring the relatively smaller profits. In such a case, we reject a job by paying a rejection penalty. Zhang and Luo [27] addressed two parallel-machine scheduling with deteriorating jobs, rejection and a fixed non-availability interval. They presented a FPTAS for the problem. Li and Yuan [12] studied parallel-machine scheduling with deteriorating jobs and rejection, the objective is to minimize the scheduling cost of the accepted jobs plus the total penalty of the rejected jobs. Gerstl and Mosheiov [2] solved a parallel identical machine scheduling problem with job-rejection and position-dependent processing times. They considered two scheduling measures: total flow-time and total load. Both problems are shown to be solved in polynomial time. Hsu and Chang [4] aimed to investigate the unrelated parallel-machine scheduling with deteriorating jobs and rejection. The objective is to minimize the sum of cost including the weighted of total load, total completion time, total absolute deviation of completion time, and the total penalty of the rejected jobs. Luo [17] presented a FPTAS for the uniform parallel-machine scheduling problem with deteriorating jobs and rejection, where the objective is to minimize the sum of the total load of the accepted jobs on all machines and the total penalties of the rejected jobs. Shabtay et al. [20] provided a detailed review of scheduling problems with rejection. Shabtay [19] studied a single machine serial batch scheduling problem with rejection. The objective is to minimize total completion time and total rejection cost. Zhao et al. [30] considered the due date assignment problem with rejection and position-dependent processing times on a single machine.

Moreover, jobs may be not ready for processing at the beginning and they arrive over time due to the limitation of supplying or storage ability. Motivated by this phenomenon, scheduling with release dates has attracted increasing attention. Liu et al. [16] considered scheduling deteriorating jobs on a single machine with release times and rejection, the objective is to minimize the makespan plus the total penalty incurred by rejecting jobs. They presented two dynamic programming algorithms and designed a FPTAS for the considered problem. Li et al. [10] studied a single machine scheduling problem with resource dependent release dates. The objective is to minimize total resource-consumption. Li et al. [13] considered parallel-batch scheduling of deteriorating jobs with release dates. The goal is to minimize the makespan. Yuan et al. [25] addressed two-agent single-machine scheduling problem with release dates and preemption. The objective is to minimize the maximum lateness. Zhang et al. [26] assessed single machine scheduling with release dates and rejection, the objective is to minimize the sum of makespan of the accepted jobs and the total rejection penalty of the rejected jobs. Liu and Luo [14] extended the study of Zhang et al. [26] to deal with single machine scheduling problem with release dates, rejection and an unavailable interval.

However, both Zhang et al. [26] and Liu and Luo [14] considered scheduling problem in which the processing times of jobs are constants. In this paper, we extend the study of Liu and Luo [14] to the case with deteriorating jobs. We consider deteriorating jobs scheduling on a single machine with simultaneous considerations of release dates, rejection and a fixed non-availability interval. The processing time of a job is a linear increasing function of its starting time. We present a FPTAS for the problem, and show that the special case without non-availability interval can be solved using the same method with a lower order.

The remainder of this paper is organized as follows: In Sect. 2 we introduce the notation and formulate the problem; In Sect. 3 we present some properties which are useful for solving the problem; In Sect. 4 we provide a FPTAS for the considered problem; In Sect. 5 we consider a special case without non-availability interval. Finally, conclusions are given in the last Section.

2 Problem formulation

The considered problem can be formulated as follows. There are a set of n jobs $\mathbf{J} = \{J_1, \ldots, J_n\}$ and a single machine with a fixed non-availability interval $[T_1, T_2]$, the machine can process at most one job at a time. The actual processing time of job J_j is an increasing function of its starting time t, that is, $p_j^A = p_j(a+bt)$, where $a \geq 0, b > 0,\ p_j > 0$ is the normal processing time of job J_j, p_j^A represents the actual processing time of job J_j. Each job has a release date r_j and a rejection penalty $w_j > 0$.

Without loss of generality, we assume that all p_j, r_j and w_j $(j = 1, 2, \ldots, n)$ are integers, and $\sum_{j=1}^n p_j^A > T_1$ which implies that not all jobs can be completed before T_1, otherwise the interval $[T_1, T_2]$ is unnecessary. Denote by A and $R = \mathbf{J} \backslash A$ the set of accepted jobs and the set of rejected jobs, respectively. Let $P_j(\pi)$ denote the completion time of the accepted job J_j in a feasible schedule π. By using the three-field notation of Graham et al. [3], the problem under consideration is denoted by

$$1, h|r_j, p_j(a+bt), rej| \underset{J_j \in A}{C_{\max}} + \sum_{J_j \in R} w_j$$

(where h denotes a fixed non-availability interval of the machine), which is NP-hard since the study of Zhang et al. [26] showed that problem $1|r_j, rej| \underset{J_j \in A}{C_{\max}} + \sum_{J_j \in R} w_j$ is NP-hard.

We present a FPTAS for the problem and then give a special case without non-availability interval denoted by $1|r_j, p_j(a+bt), rej| \underset{J_j \in A}{C_{\max}} + \sum_{J_j \in R} w_j$, which can be solved using the same method with a lower order.

3 Preliminary results

In this section, we provide some lemmas which are useful for the following results.

Lemma 1 *For the problem* $1\,|p_j(a+bt)|C_{\max}$, *if* t_0 *is the starting time of the first job, then*

$$C_{\max} = \left(t_0 + \frac{a}{b}\right) \prod\nolimits_{j=1}^n \left(1 + bp_j\right) - \frac{a}{b}.$$

Proof See Kononov and Gawiejnowicz [6]. □

Lemma 2 *The problem* $1\,|r_j, p_j(a+bt)|C_{\max}$ *is solved by sequencing the jobs in non-decreasing order of* r_j.

Proof It can be proved by a pairwise interchange of jobs. Suppose that there exists an optimal schedule $S = (\pi_1, J_j, J_i, \pi_2)$ with $r_j > r_i$, where π_1 and π_2 denote the partial sequences of S. Let S' be the schedule with jobs J_i and J_j of S mutually exchanged, that is, $S' = (\pi_1, J_i, J_j, \pi_2)$. In addition, we assume that the completion time of the last job in π_1 is B, and let $C_l(S)$ and $C_l(S')$ denote the completion times of job J_l in S and S'. We will show that the interchange of jobs J_i and J_j does not increase the objective value. The repeated implementation of this argument will lead to the optimality of the non-decreasing order of r_j for the problem $1\,|r_j,\, p_j(a+bt)|C_{\max}$. Specifically, it suffices to show that $C_j(S') \le C_i(S)$. Then we have

$$\begin{aligned}
C_j(S) &= \max\{B, r_j\} + p_j(a + b\max\{B, r_j\}) \\
&= \left(\max\{B, r_j\} + \frac{a}{b}\right)(1 + bp_j) - \frac{a}{b} \\
C_i(S) &= \max(C_j(S), r_i) + p_i(a + b\max(C_j(S), r_i)) \\
&= \left(\max\{C_j(S), r_i\} + \frac{a}{b}\right)(1 + bp_i) - \frac{a}{b} \\
&= \left(\max\left\{\left(\max\{B, r_j\} + \frac{a}{b}\right)(1 + bp_j),\ r_i + \frac{a}{b}\right\}\right)(1 + bp_i) - \frac{a}{b} \\
&= \left(\max\left\{\left(\max\{B, r_j\} + \frac{a}{b}\right)(1 + bp_j)(1 + bp_i),\ \left(r_i + \frac{a}{b}\right)(1 + bp_i)\right\}\right) - \frac{a}{b} \\
C_i(S') &= \max\{B, r_i\} + p_i(a + b\max\{B, r_i\}) \\
&= \left(\max\{B, r_i\} + \frac{a}{b}\right)(1 + bp_i) - \frac{a}{b} \\
C_j(S') &= \max(C_i(S'), r_j) + p_j(a + b\max(C_i(S'), r_j)) \\
&= \left(\max\{C_i(S'), r_j\} + \frac{a}{b}\right)(1 + bp_j) - \frac{a}{b} \\
&= \left(\max\left\{\left(\max\{B, r_i\} + \frac{a}{b}\right)(1 + bp_i),\ r_j + \frac{a}{b}\right\}\right)(1 + bp_j) - \frac{a}{b} \\
&= \left(\max\left\{\left(\max\{B, r_i\} + \frac{a}{b}\right)(1 + bp_i)(1 + bp_j),\ \left(r_j + \frac{a}{b}\right)(1 + bp_j)\right\}\right) - \frac{a}{b}
\end{aligned}$$

Since $r_j > r_i$, we have

$$\left(\max\left\{B, r_j\right\} + \frac{a}{b}\right)\left(1 + bp_j\right)(1 + bp_i) \ge \left(\max\{B, r_i\} + \frac{a}{b}\right)(1 + bp_i)\left(1 + bp_j\right) \tag{1}$$

and obviously that

$$\left(\max\left\{B, r_j\right\} + \frac{a}{b}\right)\left(1 + bp_j\right)(1 + bp_i) \ge \left(r_j + \frac{a}{b}\right)\left(1 + bp_j\right) \tag{2}$$

Then we obtain $C_j(S') \le C_i(S)$ from (1) and (2). This completes the proof of the Lemma. □

Using the similar method, we can obtain the following two Lemmas.

Lemma 3 *For the problem* $1|r_j, p_j(a+bt), rej|\, \underset{J_j \in A}{C_{\max}} + \sum_{J_j \in R} w_j$, *there exists an optimal job sequence such that the accepted jobs are sequenced in non-decreasing order of* r_j.

Lemma 4 *For the problem* $1, h|r_j, p_j(a+bt), rej|\, \underset{J_j \in A}{C_{\max}} + \sum_{J_j \in R} w_j$, *there exists an optimal job sequence such that* (i) *the accepted jobs scheduled before* T_1 *are sequenced in non-decreasing order of* r_j; (ii) *the accepted jobs scheduled after* T_2 *are sequenced in non-decreasing order of* r_j.

4 A FPTAS

Let the jobs be indexed according to $r_1 \le r_2 \le \cdots \le r_n$, We introduce variables $x_j,\ j = 1, 2, \ldots, n$, where

$$x_j = \begin{cases} 0 & \text{Job } \mathrm{J}_j \text{ is rejected.} \\ 1 & \text{Job } \mathrm{J}_j \text{ is scheduled before } \mathrm{T}_1. \\ 2 & \text{Job } \mathrm{J}_j \text{ is scheduled after } \mathrm{T}_2. \end{cases}$$

Let X be the set of all the vectors $x = (x_1, x_2, \ldots, x_n)$ with $x_j = k,\ k = 0, 1, 2$. The first j components of each vector $x \in X$ correspond to a feasible schedule of jobs $J_1, J_2, \ldots, J_j$. Let $f_j(x)$ be the objective value. $P_j^1(x)$ denotes the completion time of the last job processed before T_1. $P_j^2(x)$ represents the completion time of the last job processed after T_2. $W_j(x)$ is the total rejection penalties of jobs $J_1, J_2, \ldots, J_j$. Furthermore, if $P_j^2(x) = T_2$, then it implies that there is no job processed after T_2, thus for the first j jobs, the makespan is $P_j^1(x)$. Consequently, $f_j(x) = P_j^1(x) + W_j(x)$. While if $P_j^2(x) > T_2$, then it implies that there is at least one job processed after T_2, thus the makespan is $P_j^2(x)$, and $f_j(x) = P_j^2(x) + W_j(x)$.

Then we define the following initial and recursive functions on X :

$$\begin{aligned}
&P_0^1(x) = 0, \quad P_0^2(x) = T_2, \quad W_0(x) = 0, \quad f_0(x) = 0.\\
&P_j^k(x) = \max\{P_{j-1}^k(x), r_j\} + p_j(a + b\max\{P_{j-1}^k(x), r_j\}), \quad \text{for } x_j = k,\ k = 1, 2.\\
&P_j^i(x) = P_{j-1}^i(x), \quad \text{for } x_j = k, \quad i \ne k, \quad i = 1, 2, \quad k = 0, 1, 2.\\
&W_j(x) = W_{j-1}(x) + w_j, \quad \text{for } x_j = 0.\\
&W_j(x) = W_{j-1}(x), \quad \text{for } x_j = k, k = 1, 2.\\
&f_j(x) = P_j^1(x) + W_j(x), \quad \text{for } P_j^2(x) = T_2.\\
&f_j(x) = P_j^2(x) + W_j(x), \quad \text{for } P_j^2(x) > T_2.
\end{aligned}$$

Consequently, the problem $1, h|r_j, p_j(a+bt), rej|\ \underset{J_j \in A}{C_{\max}} + \sum_{J_j \in R} w_j$ reduces to the following problem:

$$\begin{aligned}&\text{Minimize } f_n(x)\\ &\text{Subject to } P_n^1(x) \le T_1,\ x \in X.\end{aligned}$$

We first introduce procedure Partition (A, u, δ) proposed by Kovalyov and Kubiak [7,8], where $A \subseteq X$, u is a nonnegative integer function on X, and $0 < \delta \le 1$. This procedure partitions A into disjoint subsets $A_1^u, A_2^u, \ldots, A_{k_u}^u$ such that $\left|u(x) - u(x')\right| \le \delta \min\{u(x),\ u(x')\}$ for any $x,\ x'$ from the same subset $A_j^u,\ j = 1, 2, \ldots, k_u$. The following description provides the details of partition (A, u, δ).

Procedure Partition (A, u, δ)

Step 1 Arrange vectors $x \in X$ in the order $x^{(1)}, x^{(2)}, \ldots, x^{(|A|)}$ such that $0 \le u(x^{(1)}) \le u(x^{(2)}) \le \cdots \le u(x^{(|A|)})$.

Step 2 Assign vectors $x^{(1)}, x^{(2)}, \ldots, x^{(i_1)}$ to set A_1^u until i_1 is found such that $u(x^{(i_1)}) \leq (1+\delta)u(x^{(1)})$ and $u(x^{(i_1+1)}) > (1+\delta)u(x^{(1)})$. If such i_1 does not exist, then take $A_{k_u}^u = A_1^u = A$, and stop. Assign vectors $x^{(i_1+1)}, x^{(i_1+2)}, \ldots, x^{(i_2)}$ to set A_2^u until i_2 is found such that $u(x^{(i_2)}) \leq (1+\delta)u(x^{(i_1+1)})$ and $u(x^{(i_2+1)}) > (1+\delta)u(x^{(i_1+1)})$. If such i_2 does not exist, then take $A_{k_u}^u = A_2^u = A - A_1^u$, and stop. Continue the above construction until $x^{(|A|)}$ is included in $A_{k_u}^u$ for some k_u.

Procedure Partition requires $O(|A| \log |A|)$ operations to arrange the vectors of A in nondecreasing order of $u(x)$ and $O(|A|)$ operations to provide a partition. The main properties of Partition (A, u, δ) were given in Kovalyov and Kubiak [7,8].

Property 1 $\left|u(x) - u(x')\right| \leq \delta \min\left\{u(x),\ u(x')\right\}$ for any $x, x' \in A_j^u$, $j = 1, 2, \ldots, k_u$.

Property 2 $k_u \leq \log u(x^{|A|})/\delta + 2$ for $0 < \delta \leq 1$ and $u(x^{|A|}) \geq 1$.

A formal description of the FPTAS $\mathcal{A}_\varepsilon$ for problem $1, h|r_j, p_j(a+bt), rej| \underset{J_j \in A}{C_{\max}} + \sum\limits_{J_j \in R} w_j$ is given below.

Algorithm 1 $\mathcal{A}_\varepsilon$

Step 1 (Initialization) Number the jobs so that $r_1 \leq r_2 \leq \cdots \leq r_n$. Set $Y_0 = \{(0, 0, \ldots, 0)\}$ and $j = 1$.

Step 2 (Generation of $Y_1, Y_2, \ldots, Y_n$) For set Y_{j-1}, generate set Y_j' by adding k in position j of each vector from Y_{j-1}, $k = 0, 1, 2$. Calculate the following for any $x \in Y_j'$, assuming $x_j = k$.

$$P_j^k(x) = \max\{P_{j-1}^k(x), r_j\} + p_j(a + b\max\{P_{j-1}^k(x), r_j\}),$$
$$\text{for } x_j = k,\ k = 1, 2.$$
$$P_j^i(x) = P_{j-1}^i(x), \quad \text{for } x_j = k, \quad i \neq k,\ i = 1, 2,\ k = 0, 1, 2.$$

If $P_j^1(x) > T_1$, delete x from Y_j', otherwise, go on the following computations.

$$W_j(x) = W_{j-1}(x) + w_j, \quad \text{for } k = 0,$$
$$W_j(x) = W_{j-1}(x), \quad \text{for } k = 1, 2.$$
$$f_j(x) = P_j^1(x) + W_j(x), \quad \text{for } P_j^2(x) = T_2,$$
$$f_j(x) = P_j^2(x) + W_j(x), \quad \text{for } P_j^2(x) > T_2.$$

If $j = n$, then set $Y_n = Y_n'$, and go to Step 3.
If $j < n$, then set $\delta = \varepsilon/(2(n+1))$, and perform the following computations.
Call Partition (Y_j', P_j^i, δ) $(i = 1, 2)$ to partition set Y_j' into disjoint subsets $Y_1^{P^i}, Y_2^{P^i}, \ldots, Y_{k_{Pi}}^{P^i}$.

Call Partition (Y'_j, W_j, δ) to partition set Y'_j into disjoint subsets $Y_1^W, Y_2^W, \ldots, Y_{k_W}^W$.

Call Partition (Y'_j, f_j, δ) to partition set Y'_j into disjoint subsets $Y_1^f, Y_2^f, \ldots, Y_{k_f}^f$.

Divide set Y'_j into disjoint subsets $Y_{a_1a_2bc} = Y_{a_1}^{P^1} \cap Y_{a_2}^{P^2} \cap Y_b^W \cap Y_c^f$, $a_1 = 1, 2, \ldots,$

$$k_{P^1}; \quad a_2 = 1, 2, \ldots, k_{P^2}; \quad b = 1, 2, \ldots, k_W; \quad c = 1, 2, \ldots, k_f.$$

For each nonempty subset $Y_{a_1a_2bc}$, choose a vector $x^{(a_1a_2bc)}$ such that

$$P_j^1(x^{(a_1a_2bc)}) = \min\{P_j^1(x) \,|x \in Y_{a_1a_2bc}\}$$

Set $Y_j = \{x^{(a_1a_2bc)} \,|a_1 = 1, 2, \ldots, k_{P^1}; a_2 = 1, 2, \ldots, k_{P^2}; b = 1, 2, \ldots, k_W; c = 1, 2, \ldots, k_f$ and $Y_{a_1a_2bc} = Y_{a_1}^{P^1} \cap Y_{a_2}^{P^2} \cap Y_b^W \cap Y_c^f \neq \phi\}$, and $j = j + 1$. Repeat Step 2.

Step 3 (Solution) Select vector $x^0 \in Y_n$ such that $f_n(x^0) = \min\{f_n(x) \,|x \in Y_n\}$.

For the problem $1, h|r_j, p_j(a+bt), rej|\, C_{\max}\limits_{J_j \in A} + \sum\limits_{J_j \in R} w_j$, let $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$ be an optimal solution and set $L = \log\max\{n, 1/\varepsilon, 1 + bp_{\max}, r_{\max} + \frac{a}{b}, T_2 + \frac{a}{b}, W\}$, where $p_{\max} = \max_{j=1}^n\{p_j\}$, $r_{\max} = \max_{j=1}^n\{r_j\}$, $W = \sum_{j=1}^n w_j$. Then we have the following theorem.

Theorem 1 *For the problem* $1, h|r_j, p_j(a+bt), rej|\, C_{\max}\limits_{J_j \in A} + \sum\limits_{J_j \in R} w_j$, *Algorithm* $\mathcal{A}_\varepsilon$ *finds* $x^0 \in X$ *such that* $P_n^1(x^0) \le T_1$ *and* $f_n(x^0) \le (1+\varepsilon) f_n(x^*)$ *in* $O(n^6L^5/\varepsilon^4)$.

Proof Suppose that $(x_1^*, \ldots, x_j^*, 0, \ldots, 0) \in Y_{a_1a_2bc} \subseteq Y'_j$ for some j and a_1, a_2, b, c. By the definition of $\mathcal{A}_\varepsilon$, such j always exists, for instance $j = 1$. Algorithm $\mathcal{A}_\varepsilon$ may not choose $(x_1^*, \ldots, x_j^*, 0, \ldots, 0)$ for further construction. However, a vector $x^{(a_1a_2bc)}$ such that $P_j^1(x^{(a_1a_2bc)}) = \min\{P_j^1(x) \,|x \in Y_{a_1a_2bc}\}$ is chosen instead of $(x_1^*, \ldots, x_j^*, 0, \ldots, 0)$, then we have

$$P_j^1(x^{(a_1a_2bc)}) \le P_j^1(x^*) \le T_1.$$

Since $Y_{a_1a_2bc} = Y_{a_1}^{P^1} \cap Y_{a_2}^{P^2} \cap Y_b^W \cap Y_c^f$ and $x^{(a_1a_2bc)}, (x_1^*, \ldots, x_j^*, 0, \ldots, 0) \in Y_{a_1a_2bc}$, then we have

$$x^{(a_1a_2bc)}, \left(x_1^*, \ldots, x_j^*, 0, \ldots, 0\right) \in Y_{a_1}^{P^1}, \; x^{(a_1a_2bc)}, \left(x_1^*, \ldots, x_j^*, 0, \ldots, 0\right) \in Y_{a_2}^{P^2}$$
$$x^{(a_1a_2bc)}, \left(x_1^*, \ldots, x_j^*, 0, \ldots, 0\right) \in Y_b^W, \quad x^{(a_1a_2bc)}, \left(x_1^*, \ldots, x_j^*, 0, \ldots, 0\right) \in Y_c^f.$$

□

Then according to Property 1, we have

$$\begin{aligned}
&\left|W_j(x^*) - W_j\left(x^{(a_1a_2bc)}\right)\right| \le \delta \min\left\{W_j(x^*), W_j\left(x^{(a_1a_2bc)}\right)\right\} \le \delta W_j(x^*)\\
&\left|P_j^1(x^*) - P_j^1\left(x^{(a_1a_2bc)}\right)\right| \le \delta \min\left\{P_j^1(x^*), P_j^1\left(x^{(a_1a_2bc)}\right)\right\} \le \delta P_j^1(x^*)\\
&\left|P_j^2(x^*) - P_j^2\left(x^{(a_1a_2bc)}\right)\right| \le \delta \min\left\{P_j^2(x^*), P_j^2\left(x^{(a_1a_2bc)}\right)\right\} \le \delta P_j^2(x^*)\\
&\left|f_j(x^*) - f_j\left(x^{(a_1a_2bc)}\right)\right| \le \delta \min\left\{f_j(x^*), f_j\left(x^{(a_1a_2bc)}\right)\right\} \le \delta f_j(x^*)
\end{aligned}$$

Consequently, we have

$$\begin{aligned}
&\left|W_j(x^*) - W_j\left(x^{(a_1a_2bc)}\right)\right| \le \delta W_j(x^*),\\
&\left|P_j^i(x^*) - P_j^i\left(x^{(a_1a_2bc)}\right)\right| \le \delta P_j^i(x^*), \quad i = 1, 2.\\
&\left|f_j(x^*) - f_j\left(x^{(a_1a_2bc)}\right)\right| \le \delta f_j(x^*).
\end{aligned}$$

Set $\delta = \delta_1$. We consider vector $(x_1^*, \ldots, x_j^*, x_{j+1}^*, 0, \ldots, 0)$ and $\tilde{x}^{(a_1a_2bc)} = (x_1^{(a_1a_2bc)}, \ldots, x_j^{(a_1a_2bc)}, x_{j+1}^*, 0, \ldots, 0)$. Without loss of generality, we assume $x_{j+1}^* = k$, then

$$P_{j+1}^1\left(\tilde{x}^{(a_1a_2bc)}\right) \le P_{j+1}^1(x^*) \le T_1$$

If $k = 0$, then

$$\begin{aligned}
\left|W_{j+1}(x^*) - W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= \left|W_j(x^*) + w_{j+1} - W_j\left(x^{(a_1a_2bc)}\right) - w_{j+1}\right|.\\
&\le \delta W_j(x^*) \le \delta_1 W_{j+1}(x^*)\\
\left|P_{j+1}^i(x^*) - P_{j+1}^i\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= \left|P_j^i(x^*) - P_j^i\left(x^{(a_1a_2bc)}\right)\right|\\
&\le \delta P_j^i(x^*) = \delta_1 P_{j+1}^i(x^*), \text{ for } i = 1, 2.\\
\left|f_{j+1}(x^*) - f_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= \left|f_j(x^*) + w_{j+1} - f_j\left(x^{(a_1a_2bc)}\right) - w_{j+1}\right|\\
&\le \delta f_j(x^*) \le \delta_1 f_{j+1}(x^*)
\end{aligned}$$

If $k = 1$, then

$$\begin{aligned}
\left|W_{j+1}(x^*) - W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= \left|W_j(x^*) - W_j\left(x^{(a_1a_2bc)}\right)\right|\\
&\le \delta W_j(x^*) = \delta_1 W_{j+1}(x^*)\\
\left|P_{j+1}^1(x^*) - P_{j+1}^1\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &\le \delta_1 P_{j+1}^1(x^*) \qquad (3)
\end{aligned}$$

(The proof of (3) is given in the Appendix).

$$\begin{aligned} \left|P_{j+1}^{2}(x^{*})-P_{j+1}^{2}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= \left|P_{j}^{2}(x^{*})-P_{j}^{2}\left(x^{(a_1a_2bc)}\right)\right| \\ &\leq \delta P_{j}^{2}(x^{*}) = \delta_1 P_{j+1}^{2}(x^{*}) \\ \left|f_{j+1}(x^{*})-f_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &\leq \delta_1 f_{j+1}(x^{*}) \end{aligned} \quad (4)$$

(The proof of (4) is given in the Appendix).
If $k=2$, then

$$\begin{aligned} \left|W_{j+1}(x^{*})-W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= \left|W_{j}(x^{*})-W_{j}\left(x^{(a_1a_2bc)}\right)\right| \\ &\leq \delta W_{j}(x^{*}) = \delta_1 W_{j+1}(x^{*}) \\ \left|P_{j+1}^{1}(x^{*})-P_{j+1}^{1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= \left|P_{j}^{1}(x^{*})-P_{j}^{1}\left(x^{(a_1a_2bc)}\right)\right| \\ &\leq \delta P_{j}^{1}(x^{*}) = \delta_1 P_{j+1}^{1}(x^{*}) \\ \left|P_{j+1}^{2}(x^{*})-P_{j+1}^{2}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &\leq \delta_1 P_{j+1}^{2}(x^{*}) \end{aligned} \quad (5)$$

(The proof of (5) is given in the Appendix).

$$\begin{aligned} \left|f_{j+1}(x^{*})-f_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= \Big|P_{j+1}^{2}(x^{*})+W_{j+1}(x^{*})-P_{j+1}^{2}\left(\tilde{x}^{(a_1a_2bc)}\right) \\ &\quad -W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\Big| \\ &\leq \left|P_{j+1}^{2}(x^{*})-P_{j+1}^{2}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| \\ &\quad +\left|W_{j+1}(x^{*})-W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| \\ &\leq \delta_1 P_{j+1}^{2}(x^{*})+\delta_1 W_{j+1}(x^{*}) \\ &= \delta_1 f_{j+1}(x^{*}) \end{aligned}$$

Therefore, for $x_{j+1}^{*}=k,\ k=0,1,2$, we have

$$\begin{aligned} \left|W_{j+1}(x^{*})-W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &\leq \delta_1 W_{j+1}(x^{*}), \\ \left|P_{j+1}^{i}(x^{*})-P_{j+1}^{i}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &\leq \delta_1 P_{j+1}^{i}(x^{*}), \quad i=1,2. \\ \left|f_{j+1}(x^{*})-f_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &\leq \delta_1 f_{j+1}(x^{*}). \end{aligned} \quad (6)$$

Consequently,

$$\begin{aligned} W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) &\leq (1+\delta_1)W_{j+1}(x^{*}), \\ P_{j+1}^{i}\left(\tilde{x}^{(a_1a_2bc)}\right) &\leq (1+\delta_1)P_{j+1}^{i}(x^{*}), \quad i=1,2. \end{aligned}$$

$$f_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) \le (1+\delta_1)f_{j+1}(x^*).$$

Assume that $\tilde{x}^{(a_1a_2bc)} \in Y_{c_1c_2de} \subseteq Y'_{j+1}$ and algorithm $\mathcal{A}_\varepsilon$ chooses $x^{(c_1c_2de)} \in Y_{c_1c_2de}$ instead of $\tilde{x}^{(a_1a_2bc)}$ in the $(j+1)$ st iteration. It is derived that

$$\begin{aligned}
&P^1_{j+1}\left(x^{(c_1c_2de)}\right) \le P^1_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) \le P^1_{j+1}(x^*) \le T_1\\
&\left|W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) - W_{j+1}\left(x^{(c_1c_2de)}\right)\right| \le \delta W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) \le \delta(1+\delta_1)W_{j+1}(x^*)\\
&\left|P^i_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) - P^i_{j+1}(x^{(c_1c_2de)})\right| \le \delta P^i_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) \le \delta(1+\delta_1)P^i_{j+1}(x^*)\\
&\left|f_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) - f_{j+1}\left(x^{(c_1c_2de)}\right)\right| \le \delta f_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) \le \delta(1+\delta_1)f_{j+1}(x^*)
\end{aligned} \tag{7}$$

From (6) and (7), we obtain

$$\begin{aligned}
&\left|P^i_{j+1}(x^*) - P^i_{j+1}\left(x^{(c_1c_2de)}\right)\right|\\
&\le \left|P^i_{j+1}(x^*) - P^i_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| + \left|P^i_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) - P^i_{j+1}(x^{(c_1c_2de)})\right|\\
&\le \delta_1 P^i_{j+1}(x^*) + \delta(1+\delta_1)P^i_{j+1}(x^*)\\
&= [\delta + (1+\delta)\delta_1]\, P^i_{j+1}(x^*)
\end{aligned} \tag{8}$$

Similarly,

$$\left|W_{j+1}(x^*) - W_{j+1}\left(x^{(c_1c_2de)}\right)\right| \le [\delta + (1+\delta)\delta_1]\, W_{j+1}(x^*) \tag{9}$$

$$\left|f_{j+1}(x^*) - f_{j+1}\left(x^{(c_1c_2de)}\right)\right| \le [\delta + (1+\delta)\delta_1]\, f_{j+1}(x^*) \tag{10}$$

Set $\delta_l = \delta + (1+\delta)\delta_{l-1}$, $l = 2, 3, \ldots, n-j+1$. From (8), (9) and (10), it follows

$$\begin{aligned}
\left|W_{j+1}(x^*) - W_{j+1}\left(x^{(c_1c_2de)}\right)\right| &\le \delta_2 W_{j+1}(x^*),\\
\left|P^i_{j+1}(x^*) - P^i_{j+1}\left(x^{(c_1c_2de)}\right)\right| &\le \delta_2 P^i_{j+1}(x^*), \quad i = 1, 2.\\
\left|f_{j+1}(x^*) - f_{j+1}\left(x^{(c_1c_2de)}\right)\right| &\le \delta_2 f_{j+1}(x^*).
\end{aligned}$$

By repeating the above argument for $j+2, \ldots, n$, we show that there exists $x' \in Y_n$ such that

$$\begin{aligned}
&P^1_n(x) \le P^1_n(x^*) \le T_1\\
&\left|W_n(x^*) - W_n(x')\right| \le \delta_{n-j+1} W_n(x^*),\\
&\left|P^i_n(x^*) - P^i_n(x')\right| \le \delta_{n-j+1} P^i_n(x^*), \quad i = 1, 2.\\
&\left|f_n(x^*) - f_n(x')\right| \le \delta_{n-j+1} f_n(x^*)
\end{aligned} \tag{11}$$

Since

$$\begin{aligned}\delta_{n-j+1} \leq \delta \sum_{j=0}^{n} (1+\delta)^j &= (1+\delta)^{n+1} - 1 \\ &= \sum_{k=1}^{n+1} \frac{(n+1)n \cdots (n-k+2)}{k!} \delta^k \\ &= \sum_{k=1}^{n+1} \frac{(n+1)\, n \cdots (n-k+2)}{k!\, (n+1)^k} \left(\frac{\varepsilon}{2}\right)^k \\ &\leq \sum_{k=1}^{n+1} \frac{1}{k!} \left(\frac{\varepsilon}{2}\right)^k \\ &\leq \sum_{k=1}^{n+1} \left(\frac{\varepsilon}{2}\right)^k \\ &\leq \varepsilon\end{aligned}$$

Therefore,

$$\begin{aligned}&\left|W_n(x^*) - W_n(x')\right| \leq \varepsilon W_n(x^*), \quad \left|P_n^i(x^*) - P_n^i(x')\right| \leq \varepsilon P_n^i(x^*), \quad i = 1, 2. \\ &\left|f_n(x^*) - f_n(x')\right| \leq \varepsilon f_n(x^*).\end{aligned}$$

Then in Step 3, a vector x^0 will be chosen such that

$$P_n^1(x^0) \leq P_n^1(x') \leq P_n^1(x^*) \leq T_1$$

and

$$f_n(x^0) \leq f_n(x') \leq (1+\varepsilon) f_n(x^*).$$

The time complexity of Algorithm $\mathcal{A}_\varepsilon$ can be established by noting that the most time-consuming operation of iteration j of Step 2 is a call of procedure Partition, which requires $O(\left|Y_j'\right| \log \left|Y_j'\right|)$ time to complete. To estimate $\left|Y_j'\right|$, recall that $\left|Y_{j+1}'\right| \leq 3\left|Y_j\right| \leq 3k_{P^1} \cdot k_{P^2} \cdot k_W \cdot k_f$. By Property 2, we have

$$\begin{aligned}k_{P^1} &\leq 2(n+1) \log T_1 / \varepsilon + 2 = O(nL/\varepsilon), \\ k_W &\leq 2(n+1) \log\ (W)/\varepsilon + 2 = O\left(nL/\varepsilon\right).\end{aligned}$$

And by Lemma 1, we can obtain $P_n^2(x) \leq \left[\max\{T_2, r_{\max}\} + \frac{a}{b}\right] \prod_{j=1}^{n} \left(1 + bp_j\right) - \frac{a}{b}$. Then we have

$$k_{P^2} \le 2(n+1)\log\left\{\left[\max\{T_2, r_{\max}\} + \frac{a}{b}\right]\prod\nolimits_{j=1}^{n}\left(1+bp_j\right) - \frac{a}{b}\right\}\Big/$$
$$\varepsilon + 2 = O\left(n^2L/\varepsilon\right),$$
$$k_f \le 2(n+1)\log\left\{\left[\max\{T_2, r_{\max}\} + \frac{a}{b}\right]\prod\nolimits_{j=1}^{n}\left(1+bp_j\right) - \frac{a}{b} + W\right\}\Big/$$
$$\varepsilon + 2 = O\left(n^2L/\varepsilon\right).$$

Thus $\left|Y'_j\right| = O(n^6L^4/\varepsilon^4)$ and $O(\left|Y'_j\right| \log \left|Y'_j\right|) = O(n^6L^5/\varepsilon^4)$. Therefore, Algorithm $\mathcal{A}_\varepsilon$ runs in $O(n^6L^5/\varepsilon^4)$.

5 A special case

In this section, we give the special case without non-availability interval ($T_1 = T_2 = 0$) denoted by $1|r_j, p_j(a+bt), rej|\, \underset{J_j\in A}{C_{\max}} + \sum\limits_{J_j\in R} w_j$, which can be solved using the same method with a lower order.

Similar to the problem $1, h|r_j, p_j(a+bt), rej|\, \underset{J_j\in A}{C_{\max}} + \sum\limits_{J_j\in R} w_j$, we introduce variables x_j, $j = 1, 2, \ldots, n$, where

$$x_j = \begin{cases} 0 & \text{Job } \mathrm{J_j} \text{ is rejected.} \\ 1 & \text{Job } \mathrm{J_j} \text{ is accepted.} \end{cases}$$

Let X be the set of all the vectors $x = (x_1, x_2, \ldots, x_n)$ with $x_j = k$, $k = 0, 1$. We define the following initial and recursive functions on X:

$$\begin{aligned}
&P_0(x) = 0, \quad W_0(x) = 0, \quad f_0(x) = 0.\\
&P_j(x) = P_{j-1}(x), \quad \text{for } x_j = 0.\\
&P_j(x) = \max\{P_{j-1}(x), r_j\} + p_j(a + b\max\{P_{j-1}(x), r_j\}), \quad \text{for } x_j = 1.\\
&W_j(x) = W_{j-1}(x) + w_j, \quad \text{for } x_j = 0.\\
&W_j(x) = W_{j-1}(x), \quad \text{for } x_j = 1.\\
&f_j(x) = P_j(x) + W_j(x).
\end{aligned}$$

The problem $1|r_j, p_j(a+bt), rej|\, \underset{J_j\in A}{C_{\max}} + \sum\limits_{J_j\in R} w_j$ reduces to the following problem:

$$\begin{gathered}\text{Minimize } f_n(x)\\ \text{Subject to } x \in X.\end{gathered}$$

Before giving the following algorithm $\mathcal{A}'_\varepsilon$, we first point out three major changes in this problem.

- $P_j^1(x)$ and $P_j^2(x)$ denoted in Sect. 4 are no longer necessary since the machine is available all the time in this problem. In this section, we just need define one function $P_j(x)$ as the makespan of the accepted jobs.
- In algorithm $\mathcal{A}_\varepsilon$ we need four Partitions (Y'_j, P^i_j, δ) $(i = 1, 2)$, (Y'_j, W_j, δ) and (Y'_j, f_j, δ). While in the proof of theorem 2 we will see that Partition (Y'_j, f_j, δ) is unnecessary, then in algorithm $\mathcal{A}'_\varepsilon$ we only need two Partitions (Y'_j, P_j, δ) and (Y'_j, W_j, δ).
- The constraint $P_n^1(x) \leq T_1$ is useless in this section, which means that $P_j(x)$ has no constraint. Note that in algorithm $\mathcal{A}_\varepsilon$, in order to guarantee the constraint $P_n^1(x) \leq T_1$, we must choose the vector $x^{(a_1a_2bc)}$ such that $P_j^1(x^{(a_1a_2bc)}) = \min\{P_j^1(x) \,|x \in Y_{a_1a_2bc}\}$ rather than a vector $x^{(a_1a_2bc)}$ such that $f_j(x^{(a_1a_2bc)}) = \min\{f_j(x) \,|x \in Y_{a_1a_2bc}\}$.

Algorithm 2 $\mathcal{A}'_\varepsilon$

Step 1 (Initialization) Number the jobs so that $r_1 \leq r_2 \leq \cdots \leq r_n$. Set $Y_0 = \{(0, 0, \ldots, 0)\}$ and $j = 1$.

Step 2 (Generation of $Y_1, Y_2, \ldots, Y_n$) For set Y_{j-1}, generate set Y'_j by adding k in position j of each vector from Y_{j-1}, $k = 0, 1$. Calculate the following for any $x \in Y'_j$,

$$\begin{aligned}
P_j(x) &= P_{j-1}(x), \quad \text{for } x_j = 0.\\
P_j(x) &= \max\left\{P_{j-1}(x), r_j\right\} + p_j\left(a + b\max\left\{P_{j-1}(x), r_j\right\}\right),\\
&\quad \text{for } x_j = 1.\\
W_j(x) &= W_{j-1}(x) + w_j, \text{ for } x_j = 0.\\
W_j(x) &= W_{j-1}(x), \text{ for } x_j = 1.\\
f_j(x) &= P_j(x) + W_j(x)
\end{aligned}$$

If $j = n$, then set $Y_n = Y'_n$, and go to Step 3.

If $j < n$, then set $\delta = \varepsilon/(2(n + 1))$, and perform the following computations.

Call Partition (Y'_j, P_j, δ) to partition set Y'_j into disjoint subsets $Y_1^P, Y_2^P, \ldots, Y_{k_P}^P$.

Call Partition (Y'_j, W_j, δ) to partition set Y'_j into disjoint subsets $Y_1^W, Y_2^W, \ldots, Y_{k_W}^W$.

Divide set Y'_j into disjoint subsets $Y_{ab} = Y_a^P \cap Y_b^W$, $a = 1, 2, \ldots, k_P$; $b = 1, 2, \ldots, k_W$.

For each nonempty subset Y_{ab}, choose a vector $x^{(ab)}$ such that

$$f_j(x^{(ab)}) = \min\{f_j(x) \,|x \in Y_{ab}\}$$

Set $Y_j = \{x^{(ab)} \,|a = 1, 2, \ldots, k_P\ ;\ \ b = 1, 2, \ldots, k_W$ and $Y_{ab} = Y_a^P \cap Y_b^W \neq \phi\}$, and $j = j + 1$. Repeat Step 2.

Step 3 (Solution) Select vector $x^0 \in Y_n$ such that $f_n(x^0) = \min\{f_n(x) \,|x \in Y_n\}$.

Theorem 2 *For the problem* $1|r_j, p_j(a+bt), rej|\, C_{\max}\limits_{J_j\in A} + \sum\limits_{J_j\in R} w_j$, *Algorithm* $\mathcal{A}'_\varepsilon$ *finds* $x^0 \in X$ *such that* $f_n(x^0) \le (1+\varepsilon) f_n(x^*)$ *in* $O(n^3L^3/\varepsilon^2)$.

Proof The proof is similar to that of Theorem 1, and simpler than it. We can obtain the following results from the proof of Theorem 1.

$$\left|W_n(x^*) - W_n(x')\right| \le \varepsilon W_n(x^*), \quad \left|P_n(x^*) - P_n(x')\right| \le \varepsilon P_n(x^*).$$

Since $f_j(x) = P_j(x) + W_j(x)$, then

$$\begin{aligned}\left|f_n(x^*) - f_n(x')\right| &= \left|P_n(x^*) + W_n(x^*) - P_n(x') - W_n(x')\right| \\ &\le \left|P_n(x^*) - P_n(x')\right| + \left|W_n(x^*) - W_n(x')\right| \\ &= \varepsilon P_n(x^*) + \varepsilon W_n(x^*) \\ &= \varepsilon f_n(x^*)\end{aligned}$$

So, it is derived that $f_n(x^0) \le f_n(x') \le (1+\varepsilon) f_n(x^*)$.

Similar as the time complexity of Algorithm $\mathcal{A}_\varepsilon$, to estimate $\left|Y'_j\right|$, recall that $\left|Y'_{j+1}\right| \le 2\left|Y_j\right| \le 2k_P \cdot k_W$, by Property 2, we have

$$\begin{aligned}&k_P \le 2(n+1)\log\left[\left(r_{\max} + \frac{a}{b}\right)\prod\nolimits_{j=1}^{n}\left(1+bp_j\right) - \frac{a}{b}\right]\Big/\varepsilon + 2 = O(n^2L/\varepsilon) \\ &k_W \le 2(n+1)\log(W)\big/\varepsilon + 2 = O(nL/\varepsilon).\end{aligned}$$

Thus $\left|Y'_j\right| = O(n^3L^2/\varepsilon^2)$ and $O(\left|Y'_j\right|\log\left|Y'_j\right|) = O(n^3L^3/\varepsilon^2)$. Therefore, Algorithm $\mathcal{A}'_\varepsilon$ runs in $O(n^3L^3/\varepsilon^2)$. □

6 Conclusions

In this paper, we mainly considered single machine scheduling with release dates and rejection, where the processing time of a job is a linear function of its starting time and a job can be rejected by paying a certain penalty. The machine is not available in a specified time period, and the unavailable time period is fixed and known in advance. The goal is to minimize the sum of the makespan of the accepted jobs and the total rejection penalties of the rejected jobs. We presented a FPTAS for the problem and showed the algorithm can run in $O(n^6L^5/\varepsilon^4)$, and for the special case without non-availability, we showed that can be solved in $O(n^3L^3/\varepsilon^2)$ by using the same method.

Acknowledgments We are very grateful to thank the referees for careful reading and valuable comments which led to improvements of our original manuscript.

Appendix

This appendix proves two important results which corresponding to (3) (5) and (4) in the proof of Theorem 1, respectively.

Result 1 $\left|P_{j+1}^k(x^*) - P_{j+1}^k(\tilde{x}^{(a_1a_2bc)})\right| \le \delta_1 P_{j+1}^k(x^*)$ for $x_{j+1} = k,\ k = 1, 2.$

$$\begin{aligned}
&\left|P_{j+1}^k(x^*) - P_{j+1}^k\left(\tilde{x}^{(a_1a_2bc)}\right)\right| \\
&= \Big|(1 + bp_{j+1}) \max\left\{P_j^k(x^*), r_{j+1}\right\} + ap_{j+1} \\
&\quad -(1 + bp_{j+1}) \max\left\{P_j^k\left(x^{(a_1a_2bc)}\right), r_{j+1}\right\} - ap_{j+1}\Big| \\
&= \left|(1 + bp_{j+1}) \left(\max\left\{P_j^k(x^*), r_{j+1}\right\} - \max\left\{P_j^k\left(x^{(a_1a_2bc)}\right), r_{j+1}\right\}\right)\right| \\
&= (1 + bp_{j+1}) \left|\max\left\{P_j^k(x^*), r_{j+1}\right\} - \max\left\{P_j^k\left(x^{(a_1a_2bc)}\right), r_{j+1}\right\}\right|
\end{aligned}$$

It can be divided into the following four cases, and we show that Result 1 holds for each of them.

Case 1.1 $r_{j+1} \ge P_j^k(x^*)$ and $r_{j+1} \ge P_j^k(x^{(a_1a_2bc)})$

$$\begin{aligned}
\left|P_{j+1}^k(x^*) - P_{j+1}^k\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= (1 + bp_{j+1}) \left|r_{j+1} - r_{j+1}\right| \le \delta P_{j+1}^k(x^*) \\
&= \delta_1 P_{j+1}^k(x^*).
\end{aligned}$$

Case 1.2 $P_j^k(x^{(a_1a_2bc)}) > r_{j+1} \ge P_j^k(x^*)$

$$\begin{aligned}
\left|P_{j+1}^k(x^*) - P_{j+1}^k\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= (1 + bp_{j+1}) \left|r_{j+1} - P_j^k\left(x^{(a_1a_2bc)}\right)\right| \\
&= (1 + bp_{j+1})(P_j^k\left(x^{(a_1a_2bc)}\right) - r_{j+1}) \\
&\le (1 + bp_{j+1})(P_j^k\left(x^{(a_1a_2bc)}\right) - P_j^k(x^*)) \\
&= (1 + bp_{j+1}) \left|P_j^k\left(x^{(a_1a_2bc)}\right) - P_j^k(x^*)\right| \\
&\le (1 + bp_{j+1})\delta P_j^k(x^*) \\
&\le \delta(1 + bp_{j+1}) \max\left\{P_j^k(x^*), r_{j+1}\right\} \\
&\le \delta_1 P_{j+1}^k(x^*)
\end{aligned}$$

Case 1.3 $P_j^k(x^*) > r_{j+1} \ge P_j^k(x^{(a_1a_2bc)})$

$$\begin{aligned}
\left|P_{j+1}^k(x^*) - P_{j+1}^k\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= (1 + bp_{j+1}) \left|P_j^k(x^*) - r_{j+1}\right| \\
&= (1 + bp_{j+1}) \left(P_j^k(x^*) - r_{j+1}\right)
\end{aligned}$$

$$\begin{aligned}
&\leq (1+bp_{j+1})\left(P_j^k(x^*) - P_j^k\left(x^{(a_1a_2bc)}\right)\right)\\
&= (1+bp_{j+1})\left|P_j^k(x^*) - P_j^k\left(x^{(a_1a_2bc)}\right)\right|\\
&\leq (1+bp_{j+1})\delta P_j^k(x^*)\\
&= \delta(1+bp_{j+1})\max\left\{P_j^k(x^*), r_{j+1}\right\}\\
&\leq \delta_1 P_{j+1}^k(x^*)
\end{aligned}$$

Case 1.4 $r_{j+1} < P_j^k(x^*)$ and $r_{j+1} < P_j^k(x^{(a_1a_2bc)})$

$$\begin{aligned}
\left|P_{j+1}^k(x^*) - P_{j+1}^k\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &\leq (1+bp_{j+1})\left|P_j^k(x^*) - P_j^k(x^{(a_1a_2bc)})\right|\\
&\leq (1+bp_{j+1})\delta P_j^k(x^*)\\
&= \delta(1+bp_{j+1})\max\left\{P_j^k(x^*), r_{j+1}\right\}\\
&\leq \delta_1 P_{j+1}^k(x^*)
\end{aligned}$$

Based on the above analysis, obviously we have

$$\left|P_{j+1}^k(x^*) - P_{j+1}^k\left(\tilde{x}^{(a_1a_2bc)}\right)\right| \leq \delta_1 P_{j+1}^k(x^*),\ \text{for}\quad k=1,2.$$

This completes the proof of Result 1.
Result 2 $\left|f_{j+1}(x^*) - f_{j+1}(\tilde{x}^{(a_1a_2bc)})\right| \leq \delta_1 f_{j+1}(x^*)$, for $x_{j+1}=1$.
Since $\left|f_j(x^*) - f_j(x^{(a_1a_2bc)})\right| \leq \delta f_j(x^*)$, then it implies the following four cases:
Case 2.1 $P_j^2(x^*) = T_2,\ P_j^2(x^{(a_1a_2bc)}) = T_2$

$$\begin{aligned}
\left|f_j(x^*) - f_j\left(x^{(a_1a_2bc)}\right)\right| = \Big|&P_j^1(x^*) + W_j(x^*) - P_j^1\left(x^{(a_1a_2bc)}\right)\\
&-W_j(x^{(a_1a_2bc)})\Big| \leq \delta f_j(x^*)
\end{aligned}$$

Case 2.2 $P_j^2(x^*) > T_2,\ P_j^2(x^{(a_1a_2bc)}) > T_2$

$$\begin{aligned}
\left|f_j(x^*) - f_j\left(x^{(a_1a_2bc)}\right)\right| = \Big|&P_j^2(x^*) + W_j(x^*) - P_j^2\left(x^{(a_1a_2bc)}\right)\\
&-W_j(x^{(a_1a_2bc)})\Big| \leq \delta f_j(x^*)
\end{aligned}$$

Case 2.3 $P_j^2(x^*) > T_2,\ P_j^2(x^{(a_1a_2bc)}) = T_2$

$$\begin{aligned}
\left|f_j(x^*) - f_j\left(x^{(a_1a_2bc)}\right)\right| = \Big|&P_j^2(x^*) + W_j(x^*) - P_j^1\left(x^{(a_1a_2bc)}\right)\\
&-W_j(x^{(a_1a_2bc)})\Big| \leq \delta f_j(x^*)
\end{aligned}$$

Case 2.4 $P_j^2(x^*) = T_2,\ P_j^2(x^{(a_1a_2bc)}) > T_2$

$$\begin{aligned}\left|f_j(x^*) - f_j\left(x^{(a_1a_2bc)}\right)\right| &= \left|P_j^1(x^*) + W_j(x^*) - P_j^2\left(x^{(a_1a_2bc)}\right)\right.\\ &\quad \left.-W_j(x^{(a_1a_2bc)})\right| \leq \delta f_j(x^*)\end{aligned}$$

In the following we show that Result 2 holds for each of the above four cases. We only consider Case 2.1 and Case 2.3 since Case 2.2 and Case 2.4 can be similarly proved. For Case 2.1,

$$\begin{aligned}\left|f_{j+1}(x^*) - f_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= \left|P_{j+1}^1(x^*) + W_{j+1}(x^*) - P_{j+1}^1\left(\tilde{x}^{(a_1a_2bc)}\right)\right.\\ &\quad \left.-W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right|\\ &\leq \delta_1 P_{j+1}^1(x^*) + \delta_1 W_{j+1}(x^*)\\ &= \delta_1 f_{j+1}(x^*)\end{aligned}$$

For Case 2.3,

$$\left|f_j(x^*) - f_j\left(x^{(a_1a_2bc)}\right)\right| = \left|P_j^2(x^*) + W_j(x^*) - P_j^1\left(x^{(a_1a_2bc)}\right) - W_j(x^{(a_1a_2bc)})\right|$$

If $P_j^2(x^*) + W_j(x^*) - P_j^1(x^{(a_1a_2bc)}) - W_j(x^{(a_1a_2bc)}) \geq 0$, then

$$\begin{aligned}\left|f_j(x^*) - f_j\left(x^{(a_1a_2bc)}\right)\right| &= P_j^2(x^*) - P_j^1(x^{(a_1a_2bc)}) + W_j(x^*)\\ &\quad -W_j\left(x^{(a_1a_2bc)}\right) \leq \delta f_j(x^*)\end{aligned} \tag{12}$$

Otherwise,

$$\begin{aligned}\left|f_j(x^*) - f_j\left(x^{(a_1a_2bc)}\right)\right| &= P_j^1(x^{(a_1a_2bc)}) - P_j^2(x^*) + W_j\left(x^{(a_1a_2bc)}\right)\\ &\quad -W_j(x^*) \leq \delta f_j(x^*)\end{aligned} \tag{13}$$

Either (12) or (13) is hold. Hence, we consider (12) and (13) in the following, respectively.

- If (12) is established, then we have

$$\begin{aligned}\left|f_{j+1}(x^*) - f_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= P_{j+1}^2(x^*) - P_{j+1}^1\left(\tilde{x}^{(a_1a_2bc)}\right) + W_{j+1}(x^*)\\ &\quad -W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\end{aligned} \tag{14}$$

or

$$\begin{aligned}\left|f_{j+1}(x^*) - f_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= P^1_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) - P^2_{j+1}(x^*)\\ &\quad + W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) - W_{j+1}(x^*) \qquad (15)\end{aligned}$$

Consider (14), we have

$$\begin{aligned}\left|f_{j+1}(x^*) - f_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= P^2_{j+1}(x^*) - P^1_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) + W_{j+1}(x^*)\\ &\quad - W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\\ &= P^2_j(x^*) - P^1_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) + W_j(x^*)\\ &\quad - W_j\left(x^{(a_1a_2bc)}\right)\\ &\le P^2_j(x^*) - P^1_j\left(x^{(a_1a_2bc)}\right) + W_j(x^*)\\ &\quad - W_j\left(x^{(a_1a_2bc)}\right)\\ &\le \delta f_j(x^*) \le \delta_1 f_{j+1}(x^*) \text{ (obtained from (12)).}\end{aligned}$$

Consider (15), we have

$$\begin{aligned}\left|f_{j+1}(x^*) - f_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\right| &= P^1_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) - P^2_{j+1}(x^*) + W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\\ &\quad - W_{j+1}(x^*)\\ &\le P^1_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right) - P^1_{j+1}(x^*) + W_{j+1}\left(\tilde{x}^{(a_1a_2bc)}\right)\\ &\quad - W_{j+1}(x^*)\\ &\le \delta_1 P^1_{j+1}(x^*) + \delta_1 W_{j+1}(x^*)\\ &\le \delta_1 P^2_{j+1}(x^*) + \delta_1 W_{j+1}(x^*)\\ &= \delta_1 f_{j+1}(x^*) \qquad (16)\end{aligned}$$

- While if (13) is established, then (15) must follow, and (14) is impossible.

Consider (15), we have

$$\left|f_{j+1}(x^*) - f_{j+1}(\tilde{x}^{(a_1a_2bc)})\right| \le \delta_1 f_{j+1}(x^*) \text{ (obtained from (16)).}$$

Consequently, for both (12) and (13) in Case 2.3, we can obtain

$$\left|f_{j+1}(x^*) - f_{j+1}(\tilde{x}^{(a_1a_2bc)})\right| \le \delta_1 f_{j+1}(x^*), \quad \text{for } x_{j+1} = 1.$$

Based on the analysis above, for each of the four cases we have

$$\left| f_{j+1}(x^*) - f_{j+1}(\tilde{x}^{(a_1a_2bc)}) \right| \leq \delta_1 f_{j+1}(x^*), \quad \text{for } x_{j+1} = 1.$$

This completes the proof of Result 2.

References

1. Cheng, T.C.E., Ding, Q., Lin, B.: A concise survey of scheduling with time-dependent processing times. Eur. J. Oper. Res. **152**, 1–13 (2004)
2. Gerstl, E., Mosheiov, G.: Scheduling on parallel identical machines with job-rejection and position-dependent processing times. Inf. Process. Lett. **112**, 743–747 (2012)
3. Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann. Discrete Math. **5**, 287–326 (1979)
4. Hsu, C.J., Chang, C.W.: Unrelated parallel-machine scheduling with deteriorating jobs and rejection. Appl. Mech. Mater. **263–266**, 655–659 (2013)
5. Hsu, C.J., Yang, S.J., Yang, D.L.: Due-date assignment and optional maintenance activity scheduling problem with linear deteriorating jobs. J. Mar. Sci. Technol. **19**, 97–100 (2011)
6. Kononov, A., Gawiejnowicz, S.: NP-hard cases in scheduling deteriorating jobs on dedicated machines. J. Oper. Res. Soc. **52**, 708–717 (2001)
7. Kovalyov, M.Y., Kubiak, W.: A fully polynomial approximation scheme for minimizing makespan of deteriorating jobs. J. Heuristics **3**, 287–297 (1998)
8. Kovalyov, M.Y., Kubiak, W.: A fully polynomial approximation scheme for the weighted earliness–tardiness problem. Oper. Res. **47**, 757–761 (1999)
9. Lee, C.Y.: Machine scheduling with an availability constraint. J. Global. Optim. **9**, 363–382 (1996)
10. Li, K., Yang, S.L., Ren, M.L.: Single-machine scheduling problem with resource dependent release dates to minimise total resource-consumption. Int. J. Syst. Sci. **42**, 1811–1820 (2011)
11. Li, S.S., Fan, B.Q.: Single-machine scheduling with proportionally deteriorating jobs subject to availability constraints. Asia. Pac. J. Oper. Res. **29**(4), (2012). doi:10.1142/S0217595912500194
12. Li, S.S., Y, J.J.: Parallel-machine scheduling with deteriorating jobs and rejection. Theor. Comput. Sci. **411**, 3642–3650 (2010)
13. Li, S.S., Ng, C.T., Cheng, T.C.E., Yuan, J.J.: Parallel-batch scheduling of deteriorating jobs with release dates to minimize the makespan. Eur. J. Oper. Res. **210**, 482–488 (2011)
14. Liu, C., Luo, C.X.: Single machine scheduling problem with release dates, rejection and an unavailable interval. J. Chongqing Normal Univ. **30**, 17–20 (2013)
15. Liu, J., Wang, Y., Min, X.: Single-machine scheduling with common due-window assignment for deteriorating jobs. J. Oper. Res. Soc. **65**, 291–301 (2014)
16. Liu, M., Zheng, F.F., Chu, C.B., Xu, Y.F.: Scheduling deteriorating jobs on a single machine with release times and rejection. Discrete Math. Algorithms Appl. (2012). doi:10.1142/S1793830912500322
17. Luo, C.X.: An FPTAS for uniform parallel-machine scheduling problem with deteriorating jobs and rejection. Appl. Mech. Mater. **433–435**, 2335–2338 (2013)
18. Ma, Y., Chu, C.B., Zuo, C.R.: A survey of scheduling with deterministic machine availability constraints. Comput. Ind. Eng. **58**, 199–211 (2010)
19. Shabtay, D.: The single machine serial batch scheduling problem with rejection to minimize total completion time and total rejection cost. Eur. J. Oper. Res. **233**, 64–74 (2014)
20. Shabtay, D., Gaspar, N., Kaspi, M.: A survey on offline scheduling with rejection. J. Sched. **16**, 3–28 (2013)
21. Shen, L.X., Wang, D., Wang, X.Y.: Parallel-machine scheduling with non-simultaneous machine available time. Appl. Math. Model. **37**, 5227–5232 (2013)
22. Wang, J.B., Sun, L.H., Sun, L.Y.: Single-machine total completion time scheduling with a time-dependent deterioration. Appl. Math. Model. **35**, 1506–1511 (2011)
23. Wang, X.R., Wang, J.J.: Single-machine scheduling with convex resource dependent processing times and deteriorating jobs. Appl. Math. Model. **37**, 2388–2393 (2013)
24. Wang, X.Y., Zhou, Z.L., Ji, P., Wang, J.B.: Parallel machines scheduling with simple linear job deterioration and non-simultaneous machine available times. Comput. Ind. Eng. **74**, 88–91 (2014)

25. Yuan, J.J., Ng, C.T., Cheng, T.C.E.: Two-agent single-machine scheduling with release dates and preemption to minimize the maximum lateness. J. Sched. (2013). doi:10.1007/s10951-013-0360-y
26. Zhang, L.Q., Lu, L.F., Yuan, J.J.: Single machine scheduling with release dates and rejection. Eur. J. Oper. Res. **198**, 975–978 (2009)
27. Zhang, M.J., Luo, C.X.: Parallel-machine scheduling with deteriorating jobs, rejection and a fixed non-availability interval. Appl. Math. Comput. **224**, 405–411 (2013)
28. Zhao, C.L., Tang, H.Y.: Parallel machines scheduling with deteriorating jobs and availability constraints. Jpn. J. Ind. Appl. Math. (2014). doi:10.1007/s13160-014-0150-8
29. Zhao, C.L., Ji, M., Tang, H.Y.: Parallel-machine scheduling with an availability constraint. Comput. Ind. Eng. **61**, 778–781 (2011)
30. Zhao, C.L., Yin, Y.Q., Cheng, T.C.E., Wu, C.C.: Single-machine scheduling and due date assignment with rejection and position-dependent processing times. J. Ind. Manag. Optim. **10**, 691–700 (2014)