



Algorithms for single machine scheduling problem with release dates and submodular penalties

Xiaofei Liu¹ · Man Xiao² · Weidong Li² · Yaoyu Zhu³ · Lei Ma^{4,5} 

Accepted: 4 April 2023 / Published online: 30 April 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

In this paper, we consider the single machine scheduling problem with release dates and submodular penalties, in which each job can be either assigned to the machine or rejected. The objective is to minimize the sum of the makespan of the processed jobs and the penalty of the rejected jobs which is determined by a submodular function. First, we present a simple algorithm for the off-line problem. Second, for the on-line problem, we prove that there is no on-line algorithm with a constant competitive ratio if the penalty submodular function is not monotone, and present an on-line algorithm with a competitive ratio of 3 if the penalty submodular function is monotone. Finally, we consider a special case of the on-line problem in which all jobs have the same release date. We prove that there is no on-line algorithm with a competitive ratio of $\frac{\sqrt{5}+1}{2} \approx 1.618$, and the competitive ratio of the on-line algorithm we presented is 2.

Keywords Scheduling problem · Submodular penalties · On-line algorithms · Competitive ratio

✉ Lei Ma
lei.ma@pku.edu.cn

¹ School of Information Science and Engineering, Yunnan University, Kunming, 650504, People's Republic of China

² School of Mathematics and Statistics, Yunnan University, Kunming 650504, People's Republic of China

³ School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, People's Republic of China

⁴ Beijing Academy of Artificial Intelligence, Beijing 100871, People's Republic of China

⁵ National Biomedical Imaging Center, Peking University, Beijing 100871, People's Republic of China

1 Introduction

Scheduling problem is to find a minimum scheduling such that all jobs are processed on the machines (Graham 1966). In real life, manufacturers often outsource jobs with lower processing returns to obtain greater earnings, where the outsourcing cost of a job is regarded as the penalty for rejecting this job. This problem is called scheduling problem with rejection (Bartal et al. 2000), in which each job can be rejected, but a rejection penalty is paid, and the objective is to minimize the total makespan of accepted jobs and the penalty of rejected jobs.

The relationship between the number of rejected jobs and the penalties tends to be submodular rather than linear (Zhang et al. 2018; Liu and Li 2020), where the submodular function is a set function with the property of decreasing marginal return, i.e.,

$$\pi(S) + \pi(T) \geq \pi(S \cup T) + \pi(S \cap T), \quad \forall S, T \subseteq J. \quad (1)$$

Liu and Li (2020) considered the single machine scheduling problem with release dates and submodular penalties, in which each job has a release time and the rejection penalty is determined by a submodular function

In the above problems, information of all jobs is known in advance, however, with the advent of the big data era, manufacturers must immediately and irrevocably decide to either process or reject the job when it arrives, which is denoted as on-line. For example, under the cloud computing framework, massive computing tasks generated by users are uploaded to the cloud for computing. To ensure the quality of service, the cloud must immediately decide whether to compute the task when it arrives, where the task that is not computed must pay a rejection penalty. From the perspective of maximization of profit, a cloud hopes that the tasks generated by the users should be computed as much as possible. Thus, in this paper, we consider the single machine scheduling problem with release dates and submodular penalties for the off-line case and the on-line case, respectively, where each job is either rejected or accepted and processed on the machine. The objective is to minimize the makespan of the accepted jobs plus the rejection penalty of the rejected jobs, which is determined by a submodular function.

1.1 Results and outline of the paper

In Sect. 2, we consider the off-line single machine scheduling problem with release dates and submodular penalties and present a simple algorithm.

In Sect. 3, we consider the on-line single machine scheduling problem with release dates and submodular penalties. First, we prove that there is no on-line algorithm with a constant competitive ratio if the penalty submodular function is not monotone. Second, we present an on-line algorithm with a competitive ratio of 3 if the penalty submodular function is monotone. Finally, we consider a special case of the on-line problem in which all jobs have the same release date. We prove that there is no on-line algorithm with a competitive ratio of $\frac{\sqrt{5}+1}{2} \approx 1.618$, and the competitive ratio of the

on-line algorithm we presented is 2. In Sect. 4, we conduct a simple simulation to evaluate the performance of the on-line algorithm.

We conclude the paper and suggest some possible future research in the last section.

1.2 Related work

For off-line scheduling problem, we are given n jobs and m machines, where each job has a processing time. The objective is to minimize the makespan of all jobs. Graham (1966) proved that it is strongly NP -hard, and presented the classical $(2 - 1/m)$ -approximation list scheduling (LS, for short) algorithm. Then, Graham (1969) presented a $4/3$ -approximation algorithm. Hochbaum and Shmoys (1987) presented the first polynomial time approximation scheme (PTAS). To the best of our knowledge, Jansen et al. (2020) presented the best efficient polynomial time approximation scheme (EPTAS). Jansen and Porkolab (2001) considered this problem when the number of machines is given, and presented a fully polynomial-time approximation scheme (FPTAS). Li et al. (2012) considered this problem on two machines, in which each job and machine are labeled with the grade of service levels, and presented an EPTAS.

For off-line scheduling problem with rejection, each job has a processing time and a rejection penalty. The objective is to minimize the total makespan of accepted jobs and the penalty of rejected jobs. Bartal et al. (2000) proposed a 2-approximation algorithm and a PTAS. Ou et al. (2015) proposed a $(3/2 + \epsilon)$ -approximation algorithm, where ϵ is a small given positive constant. Li et al. (2015) considered this problem under the rejection constraint, and the objective is to minimize the makespan of the accepted jobs and the total penalty of the rejected jobs is no more than a given bound. They presented an FPTAS.

For off-line scheduling problem with rejection and release data, each job has a release date, where jobs cannot be processed before their corresponding release dates. Zhang and Lu (2016) proposed a 2-approximation algorithm. Then, Zhong and Ou (2017) proposed a PTAS. When there are exactly two machines. Zhong et al. (2017) proposed a $(3/2 + \epsilon)$ -approximation algorithm. When the number of machines is one, Zhang et al. (2009) proved that it is NP -hard, and presented a 2-approximation algorithm, which was independently improved by He et al. (2016) and Ou et al. (2016).

For off-line scheduling problem with submodular penalties, the penalty is determined by a submodular function. Liu and Li (2021) considered the scheduling problem with submodular penalties, and presented a $(2 - 1/m)$ -approximation. Zhang et al. (2018) considered the precedence-constrained scheduling with submodular penalties on parallel machines, and proposed a 3-approximation algorithm. Zheng et al. (2022) and Wang and Liu (2022) independently considered the parallel-machine scheduling problem with release dates and submodular penalties, and presented a 2-approximation algorithm. Liu and Li (2020) considered the single machine scheduling problem with release dates and submodular penalties, and proposed a 2-approximation algorithm.

For on-line scheduling problem, jobs arrive one by one, and each job has to be immediately and irrevocably assigned using the arrived jobs. The performance of an on-line algorithm is measured by the competitive ratio. For on-line scheduling problem, Graham (1966) presented the LS algorithm with a competitive ratio of $2 - 1/m$.

Several algorithms have been published that have a better competitive ratio than the LS algorithm. The possible competitive ratio for on-line scheduling problem is approximately 1.92 (Albers 1999; Fleischer and Wahl 2000), whereas the best lower bound for that problem is 1.88 by (Rudin 2001). For on-line scheduling problem with rejection, Bartal et al. (2000) presented an on-line algorithm with the best-possible competitive ratio of $(\sqrt{5} + 3)/2 \approx 2.618$. For on-line scheduling problem with rejection and release data, Lu et al. (2011) provided an on-line algorithm with the best-possible competitive ratio of 2 when the number of machines is one.

2 A simple algorithm for the off-line problem

In this section, we present a simple off-line algorithm for the off-line single machine scheduling problem with release dates and submodular rejection penalties, inspired the idea in (Zhang et al. 2009).

In the off-line single machine scheduling problem with release dates and submodular rejection, we are given an instance (J, π) , where $J = \{1, 2, \dots, n\}$ is a job set, and $\pi(\cdot)$ is a submodular penalty function. Each job j , is described by the processing time p_j and the release date r_j , is either accepted and processed on the machine or rejected. The off-line problem is to choose a set $A_J \subseteq J$ of accepted jobs and reject the remaining jobs on the machine. The objective is to minimize the sum of the makespan of the accepted jobs and the penalty cost $\pi(\overline{A_J})$, where

$$\overline{A_J} = J \setminus A_J \text{ is the complement of } A_J,$$

and we assume that $\pi(\cdot)$ can be computed in polynomial time for any subset $S \subseteq J$.

If rejection is not allowed, Lawler (1973) shows that this problem can be solved using the earliest release date rule. Thus, we have the following lemma.

Lemma 2.1 (Lawler (1973)) *There exists an optimal solution such that the accepted jobs are processed using the earliest release date rule.*

Inspired the idea in (Zhang et al. 2009), our algorithm finds a job subset $A_{J,t} \subseteq J$ for each $t \in \{0, 1, 2, \dots, n\}$, and outputs the job subset with the minimum objective value, where t represents the maximum index of the job to be processed in the job subset. Thus, for any $t \in \{0, 1, 2, \dots, n\}$, jobs in $B_t = \{j \in J \mid j > t\}$ are all rejected, and only jobs in $J \setminus B_t$ need to be considered for being either processed or rejected. To determine the $A_{J,t}$, an auxiliary function $p\pi_t(\cdot)$ defined on all the subset of $J \setminus B_t$ is constructed as follows.

$$p\pi_t(S) = p(\overline{S \cup B_t}) + \pi(S \cup B_t), \quad \forall S \subseteq J \setminus B_t,$$

where we define

$$p(J') = \sum_{j:j \in J'} p_j, \quad \forall J' \subseteq J.$$

Lemma 2.2 $p\pi_t(\cdot)$ is a submodular function.

Proof Since $B_t = \{j \in J \mid j > t\}$, for any two job sets $S_1, S_2 \subseteq J \setminus B_t$, we have

$$\begin{aligned} & p\pi_t(S_1) + p\pi_t(S_2) \\ &= p(\overline{S_1 \cup B_t}) + \pi(S_1 \cup B_t) + p(\overline{S_2 \cup B_t}) + \pi(S_2 \cup B_t) \\ &= p(\overline{(S_1 \cup S_2) \cup B_t}) + p(\overline{(S_1 \cap S_2) \cup B_t}) + \pi(S_1 \cup B_t) + \pi(S_2 \cup B_t) \\ &\geq p(\overline{(S_1 \cup S_2) \cup B_t}) + p(\overline{(S_1 \cap S_2) \cup B_t}) + \pi((S_1 \cup S_2) \cup B_t) + \pi((S_1 \cap S_2) \cup B_t) \\ &= p\pi_t(S_1 \cup S_2) + p\pi_t(S_1 \cap S_2), \end{aligned}$$

where the inequality follows from the fact that $\pi(\cdot)$ is a submodular function. Thus, $p\pi_t(\cdot)$ is a submodular function. \square

Based on Lemma 2.2 and the algorithm designed in Iwata and Orlin (2009),

$$S_t := \arg \min_{S: S \subseteq J \setminus B_t} p\pi_t(S)$$

i.e.,

$$p\pi_t(S_t) \leq p\pi_t(S), \quad \forall S \subseteq J \setminus B_t. \tag{2}$$

can be found in polynomial-time. Then, we defined $A_{J,t} = J \setminus (S_t \cup B_t) = \overline{S_t \cup B_t}$, and the objective value is $Z_{J,t} = r_t + p(A_{J,t}) + \pi(\overline{A_{J,t}})$, where $r_0 = 0$. Output $A_J = \arg \min_{A_{J,t}} Z_{J,t}$. We provide the detailed the simple off-line algorithm in Algorithm 1.

Algorithm 1: off-line algorithm

- 1 Initially, let $r_0 = 0$.
 - 2 **for** $t \in \{0, 1, 2, \dots, n\}$ **do**
 - 3 Set $B_t = \{j \in J \mid j > t\}$ and construct the auxiliary function $p\pi_t(\cdot)$ as above.
 - 4 Using the algorithm proposed by Iwata and Orlin (2009), find $S_t := \arg \min_{S: S \subseteq J \setminus B_t} p\pi_t(S)$.
 - 5 Set $A_{J,t} := \overline{S_t \cup B_t}$ and $Z_{J,t} := r_t + p(A_{J,t}) + \pi(\overline{A_{J,t}})$.
 - 6 Let $A_J := \arg \min_{A_t} Z_{J,t}$, and output A_J and its objective value Z_J .
-

Let Z^* be the optimal value. We have the following lemma.

Theorem 2.3 $Z_J \leq 2Z^*$.

Proof Let A^* be an optimal solution and its objective value is Z^* . If the optimal solution rejects all the jobs, $A^* = \emptyset$, then $A_{J,0}$ is exactly the optimal solution, which implies that the theorem holds; otherwise, let $t_{A^*} = \max\{j \mid j \in A^*\}$ be the maximum index of the job in A^* , and

$$r_{t_{A^*}} + \pi(\overline{A^*}) \leq C(A^*) + \pi(\overline{A^*}) = Z^*, \tag{3}$$

where $C(A^*)$ is the makespan of the jobs in A^* using the earliest release date rule.

Since $A_{J,t_{A^*}} = \overline{S_{t_{A^*}} \cup B_{t_{A^*}}}$, we have

$$\begin{aligned}
 p(A_{J,t_{A^*}}) + \pi(\overline{A_{J,t_{A^*}}}) &= p(\overline{S_{t_{A^*}} \cup B_{t_{A^*}}}) + \pi(S_{t_{A^*}} \cup B_{t_{A^*}}) \\
 &= p\pi_{t_{A^*}}(S_{t_{A^*}}) \\
 &\leq p\pi_{t_{A^*}}(\overline{A^*} \setminus B_{t_{A^*}}) \\
 &= p((\overline{A^*} \setminus B_{t_{A^*}}) \cup B_{t_{A^*}}) + \pi((\overline{A^*} \setminus B_{t_{A^*}}) \cup B_{t_{A^*}}) \\
 &= p(A^*) + \pi(\overline{A^*}) \\
 &\leq Z^*,
 \end{aligned}
 \tag{4}$$

where the fourth equality follows from $B_{t_{A^*}} = \{j \in J \mid j > t_{A^*}\} \subseteq \overline{A^*}$, the first inequality follows from inequality (2), and the second inequality follows from $p(A^*) \leq C(A^*)$.

Thus, the objective value of A_J generated by Algorithm 1 is

$$\begin{aligned}
 Z_J &\leq Z_{J,t_{A^*}} \\
 &= r_{t_{A^*}} + p(A_{J,t_{A^*}}) + \pi(\overline{A_{J,t_{A^*}}}) \\
 &\leq 2Z^*,
 \end{aligned}$$

where the first inequality follows from $A_{J,t} = \arg \min_{A_{J,t}} Z_{J,t}$, and the second inequality follows from inequalities (3) and (4). □

Next, we use an example to illustrate our algorithm and analysis of Theorem 2.3 is tight:

Example 1 we are given an instance (J, π) , where $J = \{1, 2, \dots, n\}$; $r_j = 0$ and $p_j = 1$ for $j \in \{1, 2, \dots, n - 1\}$; $r_n = n - 1$ and $p_n = 1$; and the submodular function $\pi(\cdot)$ is defined as follows:

$$\left\{ \begin{array}{l} \pi(\emptyset) = 0; \\ \pi(\{n\}) = 2n - 1; \\ \pi(S) = \sum_{j:j \in S} (1 + \frac{1}{n}), \text{ if } n \notin S; \\ \pi(S) = 2n - 1, \text{ otherwise.} \end{array} \right.$$

It is easy to obtain that the optimal solution is to process all jobs, and the optimal value is $Z^* = n$.

Let A_J be the output job set generated by Algorithm 1, and $A_J = \arg \min_{A_{J,t}} Z_{J,t}$, where $Z_{J,t}$ is the objective value of $A_{J,t}$. When $t \in \{0, 1, 2, \dots, n - 1\}$, job n is in B_t and $n \notin A_{J,t}$. This means, $A_{J,t}$ is empty set and its objective value is $Z_{J,t} = r_t + p(A_{J,t}) + \pi(\overline{A_{J,t}}) = 0 + p(\emptyset) + \pi(J) = 2n$; When $t = n$, $A_{J,t}$ contains all jobs in J and the objective value of $A_{J,t}$ is $Z_{J,t} = r_n + p(A_{J,t}) + \pi(\overline{A_{J,t}}) =$

$n - 1 + p(J) + \pi(\emptyset) = 2n - 1$. This means, $A_J = A_{J,n} = J$, and its objective is $Z = Z_{J,n} = 2n - 1$. Thus, we have

$$\frac{Z}{Z^*} = \frac{2n - 1}{n + 1} \rightarrow 2, \text{ when } n \rightarrow \infty.$$

3 The on-line problem

In this section, we consider the on-line single machine scheduling problem with release dates and submodular rejection penalties. First, we provide a formal problem statement, and proved that the lower bound of this problem. Second, we present an on-line algorithm. Finally, we consider a special case of this problem.

3.1 Low bound

We are given a single machine and a sequence of n jobs, $J = \{1, 2, \dots, n\}$, arriving online and a submodular penalty function $\pi(\cdot)$, which is given as a value oracle. Each job j in J , is described by the processing time p_j and the release date r_j , where we assume that

$$r_j \leq r_{j'} \text{ for any } 1 \leq j < j' \leq n.$$

The job is to be either scheduled or rejected immediately and irrevocably at the time of it arrivals. The on-line problem is to find a job subset $A^J \subseteq J$. The objective is to minimize the sum of the makespan of the accepted jobs and the penalty cost $\pi(\overline{A^J})$, where

$$\overline{A^J} = J \setminus A^J \text{ is the complement of } A^J,$$

Theorem 3.1 *There is no on-line algorithm with a constant competitive ratio for the on-line problem if the penalty submodular function $\pi(\cdot)$ is not monotone, where monotone function satisfies $\pi(S) \leq \pi(T)$ for any $S \subseteq T \subseteq J$.*

Proof We are given a sequence of two jobs, $J = \{1, 2\}$, arriving on-line, and a penalty submodular function $\pi(\cdot)$ defined as follows,

$$\begin{cases} \pi(\emptyset) = 0; \\ \pi(\{1\}) = P^2 + 1; \\ \pi(\{2\}) = 1; \\ \pi(\{1, 2\}) = 1. \end{cases}$$

Job 1 arrives, where $r_1 = 0$ and $p_1 = P$. For any algorithm \mathcal{A} , If \mathcal{A} rejects job 1, then no job arrives, and we have $Z_{\mathcal{A}} = \pi(\{1\}) = P^2 + 1$ and $Z^* = r_1 + p_1 = P$, where

$Z_{\mathcal{A}}$ is the objective value of job set generated by \mathcal{A} and Z^* is the optimal value. This implies that

$$\frac{Z_{\mathcal{A}}}{Z^*} = \frac{P^2 + 1}{P} > P.$$

If \mathcal{A} processes job 1 on the machine, then job 2 arrives, where $r_2 = 1$ and $p_2 = P$. The optimal solution is to reject all jobs and its objective value is $Z^* = \pi(\{1, 2\}) = 1$. If \mathcal{A} rejects job 2, we have $Z_{\mathcal{A}} = 0 + P + \pi(\{2\}) = P + 1$; otherwise \mathcal{A} processes job 2 on the machine, we have $Z_{\mathcal{A}} = 0 + P + P = 2P$. These statements imply that

$$\frac{Z_{\mathcal{A}}}{Z^*} \geq \frac{P + 1}{1} > P.$$

Since P is any positive number, the lemma holds. □

Therefore, in the following part of this section, we assume that the penalty sub-modular function is monotone.

3.2 An algorithm for the on-line problem

Then, we present an on-line algorithm based on the off-line algorithm. For any $l \in \{1, 2, \dots, n\}$, let $J_l = \{1, \dots, l\}$ be the set of the first l jobs in J . When the job l arrives, all information of the jobs in J_l is revealed. Thus, we can find a job set A_{J_l} for instance (J_l, π) using the Algorithm 1.

Further analysis of the relationship between A_{J_l} and job l , we can get the following lemmas. Let $t_{A_{J_l}} = \max\{j | j \in A_{J_l}\}$.

Lemma 3.2 *For any $t \in \{0, 1, 2, \dots, l\}$ and any $S \subseteq J_l$ with $B_t = \{j \in J_l | j > t\} \subseteq S$, we have*

$$r_{t_{A_{J_l}}} + p(A_{J_l}) + \pi(\overline{A_{J_l}}) \leq r_t + p(\overline{S}) + \pi(S).$$

In particular, for any $S \subseteq J$, we have

$$r_{t_{A_{J_l}}} + p(A_{J_l}) + \pi(\overline{A_{J_l}}) \leq r_l + p(\overline{S}) + \pi(S).$$

Proof Given an integer $t \in \{0, 1, 2, \dots, l\}$, for any $S \subseteq J_l$ with $B_t \subseteq S$, since $A_{J_l,t} = \overline{S}_t \cup B_t$, we have

$$\begin{aligned} p(A_{J_l,t}) + \pi(\overline{A_{J_l,t}}) &= p(\overline{S}_t \cup B_t) + \pi(S_t \cup B_t) \\ &= p\pi_t(S_t) \\ &\leq p\pi_t(S \setminus B_t) \\ &= p(\overline{(S \setminus B_t) \cup B_t}) + \pi((S \setminus B_t) \cup B_t) \\ &= p(\overline{S}) + \pi(S), \end{aligned}$$

where S_t and $A_{J_l,t}$ are the job sets generated by the off-line algorithm for instance (J_l, π) , the last equality follows from $B_t \subseteq S$, and the inequality follows from inequality (2). Since $A_{J_l} = \arg \min_{A_{J_l,t}} Z_{J_l,t}$, the objective value of A_{J_l} is

$$r_{t_{A_{J_l}}} + p(A_{J_l}) + \pi(\overline{A_{J_l}}) \leq r_t + p(A_{J_l,t}) + \pi(\overline{A_{J_l,t}}) \leq r_t + p(\overline{S}) + \pi(S),$$

for any set S with $B_t \subseteq S$. Since $B_l = \emptyset$ and

$$r_{t_{A_{J_l}}} + p(A_{J_l}) + \pi(\overline{A_{J_l}}) \leq r_l + p(\overline{S}) + \pi(S), \quad \forall S \subseteq J.$$

□

Lemma 3.3 *There exists an optimal solution $A_{J_l}^*$ satisfying $l \in A_{J_l}^*$ when $l \in A_{J_l}$.*

Proof Assume that any optimal solution satisfying that job l is rejected, i.e., $l \notin A_{J_l}^*$, where $A_{J_l}^*$ is an optimal solution. Since $l \in A_{J_l}$, then $A_{J_l}^* \cup A_{J_l}$ is not an optimal solution, and

$$\begin{aligned} Z_{J_l}^* &= C(A_{J_l}^*) + \pi(\overline{A_{J_l}^*}) \\ &< C(A_{J_l}^* \cup A_{J_l}) + \pi(\overline{A_{J_l}^* \cup A_{J_l}}) \\ &\leq \max\{C(A_{J_l}^*), r_l\} + p(A_{J_l} \setminus A_{J_l}^*) + \pi(\overline{A_{J_l}^* \cup A_{J_l}}), \end{aligned} \tag{5}$$

where $Z_{J_l}^*$ is the optimal value for instance (J_l, π) , and $C(A^*)$ is the makespan of the jobs in $A_{J_l}^*$ using the earliest release date rule.

Case 1. if $r_l \leq C(A_{J_l}^*)$, by rearranging inequality (5), we have

$$\begin{aligned} p(A_{J_l} \setminus A_{J_l}^*) &> Z_{J_l}^* - C(A_{J_l}^*) - \pi(\overline{A_{J_l}^* \cup A_{J_l}}) \\ &= \pi(\overline{A_{J_l}^*}) - \pi(\overline{A_{J_l}^* \cup A_{J_l}}) \\ &\geq \pi(\overline{A_{J_l}^* \cap A_{J_l}}) - \pi(\overline{A_{J_l}}). \end{aligned} \tag{6}$$

where the second equality follows from inequality (1). Then,

$$\begin{aligned} p(A_{J_l}) + \pi(\overline{A_{J_l}}) &= p(A_{J_l} \cap A_{J_l}^*) + p(A_{J_l} \setminus A_{J_l}^*) + \pi(\overline{A_{J_l}}) \\ &> p(A_{J_l} \cap A_{J_l}^*) + \pi(\overline{A_{J_l}^* \cap A_{J_l}}), \end{aligned}$$

where the inequality follows from inequality (6). This statement and $r_l \geq r_{t_{A_{J_l}^*}}$ imply that

$$r_l + p(A_{J_l}) + \pi(\overline{A_{J_l}}) > r_{t_{A_{J_l}^*}} + p(A_{J_l} \cap A_{J_l}^*) + \pi(\overline{A_{J_l}^* \cap A_{J_l}}),$$

which contradicts Lemma 3.2 by $B_{t_{A_{J_l}^*}} \subseteq \overline{A_{J_l}^* \cap A_{J_l}}$ and $t_{A_{J_l}} = \max\{j | j \in A_{J_l}\} = l$, where $t_{A_{J_l}^*} = \max\{j | j \in A_{J_l}^*\}$.

Case 2. if $r_l > C(A_{J_l}^*)$, let $t_{A_{J_l}^*} = \max\{j | j \in A_{J_l}^*\}$, then

$$r_{t_{A_{J_l}^*}} \leq C(A_{J_l}^*) < r_l$$

and

$$Z_{J_l}^* \geq r_{t_{A_{J_l}^*}} + \pi(\overline{A_{J_l}^*}) \text{ and } Z_{J_l}^* < r_l + p(A_{J_l} \setminus A_{J_l}^*) + \pi(\overline{A_{J_l}^* \cup A_{J_l}}), \tag{7}$$

where the first inequality follows from inequality (3), and the second inequality follows from inequality (5). Thus, we have

$$\begin{aligned} r_l - r_{t_{A_{J_l}^*}} &> Z_{J_l}^* - p(A_{J_l} \setminus A_{J_l}^*) - \pi(\overline{A_{J_l}^* \cup A_{J_l}}) - (Z_{J_l}^* - \pi(\overline{A_{J_l}^*})) \\ &= \pi(\overline{A_{J_l}^*}) - \pi(\overline{A_{J_l}^* \cup A_{J_l}}) - p(A_{J_l} \setminus A_{J_l}^*) \\ &\geq \pi(\overline{A_{J_l}^* \cap A_{J_l}}) - \pi(\overline{A_{J_l}}) - p(A_{J_l} \setminus A_{J_l}^*) \\ &= \pi(\overline{A_{J_l}^* \cap A_{J_l}}) - \pi(\overline{A_{J_l}}) - (p(A_{J_l}) - p(A_{J_l} \cap A_{J_l}^*)) \\ &= p(A_{J_l} \cap A_{J_l}^*) + \pi(\overline{A_{J_l}^* \cap A_{J_l}}) - (p(A_{J_l}) + \pi(\overline{A_{J_l}})), \end{aligned}$$

where the first inequality follows from inequality (7), and the second inequality follows from inequality (1). This implies that $r_l + p(A_{J_l}) + \pi(\overline{A_{J_l}}) > r_{t_{A_{J_l}^*}} + p(A_{J_l} \cap A_{J_l}^*) + \pi(\overline{A_{J_l}^* \cap A_{J_l}})$, which contradicts Lemma 3.2.

Thus, the lemma holds. □

Then, we introduce the on-line algorithm. When a new job j is arrives, the algorithm works by assigning job j based on the job set A_{J_j} generated by Algorithm 1 for instance (J_j, π) . If $j \in A_{J_j}$, job j is processed on the machine; otherwise, job j is rejected. Then, we provide the detailed the on-line algorithm in Algorithm 2.

Algorithm 2:

- 1 Initially, set the makspan $C = 0$, $j = 1$, $A = \emptyset$ and $R = \emptyset$.
 - 2 Assume that a job j arrives.
 - 3 Using Algorithm 1, let A_{J_j} be the output job set for instance (J_j, π) .
 - 4 **if** $j \in A_{J_j}$ **then**
 - 5 Process job j on the machine at time $\max\{C, r_j\}$, and set $C := \max\{C, r_j\} + p_j$ and $A := A \cup \{j\}$.
 - 6 If no new job arrives, stop and output the current job set A and its value $Z = C + \pi(\overline{A})$; otherwise, set $j := j + 1$ and go to 2.
-

For any $j \in \{1, 2, \dots, n\}$, let $Z_{J_j}^*$ be the optimal value for instance (J_j, π) of the off-line case, and let C^j be the makespan when job j is assigned by Algorithm 2.

Lemma 3.4 $C^j \leq Z_{J_j}^*, \forall j \in \{1, 2, \dots, n\}$.

Proof Our proof is by mathematical induction. When $j = 1$, if $A_{J_1} = \{1\}$, then job 1 is processed on the machine by Algorithm 2, and $C^1 = r_1 + p_1 = Z_{J_1}^*$ by Lemma 3.3; otherwise, then job 1 is rejected by Algorithm 2, and $C^1 = 0 \leq Z_{J_1}^*$. Therefore, we have $C^1 \leq Z_{J_1}^*$.

We prove that $C^l \leq Z_{J_l}^*$ if $C^{l-1} \leq Z_{J_{l-1}}^*$ for any integer $l > 1$, and show this by proving the following two cases.

Case 1. if $l \notin A_{J_l}$, then job l is rejected by Algorithm 2, and

$$C^l = C^{l-1} \leq Z_{J_{l-1}}^* \leq Z_{J_l}^*.$$

Case 2. if $l \in A_{J_l}$, then job l is processed on the machine by Algorithm 2, and

$$C^l = \max\{C^{l-1}, r_l\} + p_l.$$

Based on Lemma 3.3, there exists an optimal solution $A_{J_l}^*$ for instance (J_l, π) satisfying $l \in A_{J_l}^*$, and the objective value of $A_{J_l}^*$ is

$$\begin{aligned} Z_{J_l}^* &= C(A_{J_l}^*) + \pi(\overline{A_{J_l}^*}) \\ &= \max\{C(A_{J_l}^* \setminus \{l\}), r_l\} + p_l + \pi(\overline{A_{J_l}^*}) \\ &= \max\{C(A_{J_l}^* \setminus \{l\}) + \pi(\overline{A_{J_l}^*}), r_l + \pi(\overline{A_{J_l}^*})\} + p_l \\ &\geq \max\{Z_{J_{l-1}}^*, r_l + \pi(\overline{A_{J_l}^*})\} + p_l \\ &\geq \max\{C^{l-1}, r_l\} + p_l \\ &= C^l, \end{aligned} \tag{8}$$

where $C(A_{J_l}^*)$ is the makespan of job set $A_{J_l}^*$ using the earliest release date rule; the first inequality follows from the fact that $A_{J_l}^* \setminus \{l\}$ is a feasible job set for instance (J_{l-1}, π) and $Z_{J_{l-1}}^*$ is the optimal value for instance (J_{l-1}, π) ; and the second inequality follows from $C^{l-1} \leq Z_{J_{l-1}}^*$ and $\pi(S) \geq 0$ for any $S \subseteq J$. □

When a new job j arrives, let A^j be the processed job set when job j is assigned by Algorithm 2. For convenience, we define

$$R^j = \overline{A^j}.$$

Lemma 3.5 $\pi(R^j) \leq 2Z_{J_j}^*, \forall j \in \{1, 2, \dots, n\}$.

Proof For instance (J_j, π) , let A_{J_j} be the job set generated by Algorithm 1, and its objective is

$$r_{t_{A_{J_j}}} + p(A_{J_j}) + \pi(\overline{A_{J_j}}) \leq 2Z_{J_j}^*, \tag{9}$$

where $t_{A_{J_j}} = \max\{j | j \in A_{J_j}\}$ and the inequality follows from Theorem 2.3.

Case 1. if $R^j \cap A_{J_j} = \emptyset$, we have $R^j \subseteq \overline{A_{J_j}}$ and

$$\pi(R^j) \leq \pi(\overline{A_{J_j}}) \leq 2Z_{J_j}^*,$$

where the first inequality follows from that $\pi(\cdot)$ is monotone, and the second inequality follows from inequality (9).

Case 2. if $R^j \cap A_{J_j} \neq \emptyset$, let l be the job with the maximum index in $R^j \cap A_{J_j}$, then we have

$$r_l \leq r_{t_{A_{J_j}}} \tag{10}$$

by $t_{A_{J_j}} = \max\{j | j \in A_{J_j}\}$. Let A_{J_l} be the job set for instance (J_l, π) using Algorithm 1. Then, we prove the following inequalities in instance (J_l, π) , and use

$$R_{J_j} = J_j \setminus A_{J_j} \text{ instead of the } \overline{A_{J_j}} \text{ we used before.}$$

In the the following proof, we define

$$\overline{S} = J_l \setminus S \text{ for any } S \subseteq J_l.$$

Since job $l \in R^j$ is rejected by Algorithm 2, we have $l \notin A_{J_j}$. Based on Lemma 3.2, we have $r_{t_{A_{J_l}}} + p(A_{J_l}) + \pi(\overline{A_{J_l}}) \leq r_l + p(\overline{S}) + \pi(S)$ for any $S \subseteq J_l$, where $t_{A_{J_l}} = \max\{j | j \in A_{J_l}\}$. By rearranging the terms and let $S = \overline{A_{J_l}} \cap R_{J_j}$, we have

$$\begin{aligned} r_l - r_{t_{A_{J_l}}} &\geq p(A_{J_l}) + \pi(\overline{A_{J_l}}) - \left(p(\overline{\overline{A_{J_l}} \cap R_{J_j}}) + \pi(\overline{A_{J_l}} \cap R_{J_j}) \right) \\ &= p(A_{J_l}) - p(A_{J_l} \cup (J_l \setminus R_{J_j})) + \pi(\overline{A_{J_l}}) - \pi(\overline{A_{J_l}} \cap R_{J_j}) \\ &= -p((J_l \setminus R_{J_j}) \setminus A_{J_l}) + \pi(\overline{A_{J_l}}) - \pi(\overline{A_{J_l}} \cap R_{J_j}) \\ &\geq -p(A_{J_j} \setminus A_{J_l}) + \pi(\overline{A_{J_l}} \cup R_{J_j}) - \pi(R_{J_j}) \end{aligned}$$

where the last inequality follows from inequality (1) and $J_l \setminus R_{J_j} \subseteq J_j \setminus R_{J_j} = A_{J_j}$. By rearranging the terms, we have

$$\begin{aligned} \pi(\overline{A_{J_l}} \cup R_{J_j}) &\leq p(A_{J_j} \setminus A_{J_l}) + \pi(R_{J_j}) + r^l - r_{t_{A_{J_l}}} \\ &\leq r^l + p(A_{J_j} \setminus A_{J_l}) + \pi(R_{J_j}). \end{aligned} \tag{11}$$

If $R^j \cap (A_{J_j} \setminus \overline{A_{J_l}}) = \emptyset$, then we have

$$R^j \subseteq J_j \setminus (A_{J_j} \setminus \overline{A_{J_l}}) = R_{J_j} \cup (A_{J_j} \cap \overline{A_{J_l}}) \subseteq \overline{A_{J_l}} \cup R_{J_j},$$

and

$$\pi(R^j) \leq \pi(\overline{A_{J_l}} \cup R_{J_j}) \leq r^l + p(A_{J_j} \setminus A_{J_l}) + \pi(R_{J_j}) \leq r_{t_{A_{J_j}}} + p(A_j) + \pi(R_j) \leq 2Z_{J_j}^*,$$

where the first inequality follows from that $\pi(\cdot)$ is monotone, and the second inequality follows from inequality (11) and inequality (10); otherwise, let $A_{J_j} = A_{J_j} \setminus \overline{A_{J_l}}$, and repeat the above **Case 2**. Repeating at most n times, we can obtain that the equality $R^j \cap (A_{J_j} \setminus \overline{A_{J_l}}) = \emptyset$ follows. \square

By Lemma 3.4 and Lemma 3.5, the following theorem is obvious.

Theorem 3.6 *The competitive ratio of Algorithm 2 for the on-line problem is 3.*

We complement this result by an example implying that the analysis of Theorem 3.6 is tight, and consider the following instance with n jobs, where $r_j = 0$ and $p_j = 1$ for $j \in \{1, 2, \dots, n - 1\}$, and $r_n = n - 1$ and $p_n = 1$. The submodular function $\pi(\cdot)$ is a value oracle which is defined as follows (same to Example 1):

$$\begin{cases} \pi(\emptyset) = 0; \\ \pi(\{n\}) = 2n - 1; \\ \pi(S) = \sum_{j:j \in S} (1 + \frac{1}{n}), \text{ if } n \notin S; \\ \pi(S) = 2n - 1, \text{ otherwise.} \end{cases}$$

Same to Example 1, the optimal solution is to process all jobs, and the optimal value is $Z^* = n$.

For any $S \subseteq \{1, 2, \dots, n - 1\}$, we have $p(S) < \pi(S)$. For $j \in \{1, 2, \dots, n - 1\}$, since $r_j = 0$, the output job set contains all jobs for instance (J_j, π) using Algorithm 1. i.e., $A_{J_j} = J_j$. This means, the output job set by Algorithm 2 is to contain the first n jobs.

For instance (J_n, π) , same as Example 1, A_{J_n} dose not contain job n , and the output job set by Algorithm 2 is not contains job n , i.e., the objective value of A^j is $Z = C(J_{n-1}) + \pi(\{n\}) = n - 1 + 2n - 1 = 3n - 2$. Thus, we have

$$\frac{Z}{Z^*} = \frac{3n - 2}{n} \rightarrow 3, \text{ when } n \rightarrow \infty.$$

3.3 A special case with the same release date

Then, we consider the special case of the on-line problem, where all the jobs have the same release date.

Theorem 3.7 *There is no on-line algorithm with a competitive ratio less than $\frac{\sqrt{5}+1}{2}$, even when $r_j = 0$ for any $j \in J$.*

Proof We are given a sequence of two jobs, $J = \{1, 2\}$, arriving on-line, and a penalty submodular function $\pi(\cdot)$ defined as follows,

$$\begin{cases} \pi(\emptyset) = 0; \\ \pi(\{1\}) = \pi(\{2\}) = \pi(\{1, 2\}) = \frac{\sqrt{5} + 1}{2}. \end{cases}$$

Job 1 arrives, where $p_j = 1$. For any algorithm \mathcal{A} , if job 1 is rejected by \mathcal{A} , then no job arrives, and $Z_{\mathcal{A}} = \pi(\{1\}) = \frac{\sqrt{5}+1}{2}$ and $Z^* = r_j + p_1 = 1$, where $Z_{\mathcal{A}}$ is the objective value of the job set generated by \mathcal{A} and Z^* is the optimal value. This implies that

$$\frac{Z_{\mathcal{A}}}{Z^*} \geq \frac{\sqrt{5} + 1}{2}.$$

If job 1 is processed on the machine by \mathcal{A} , then job 2 arrives, where $p_2 = 10$. The optimal solution is to reject all jobs and its optimal value is $Z^* = \pi(\{1, 2\}) = \frac{\sqrt{5}+1}{2}$. If job 2 is rejected by \mathcal{A} , then $Z_{\mathcal{A}} = 0 + 1 + \pi(\{2\}) = 1 + \frac{\sqrt{5}+1}{2} = \frac{\sqrt{5}+3}{2}$. Otherwise, job 2 is processed on the machine, and $Z_{\mathcal{A}} = p_1 + p_2 = 11$. These statements imply that

$$\frac{Z_{\mathcal{A}}}{Z^*} \geq \frac{\frac{\sqrt{5}+3}{2}}{\frac{\sqrt{5}+1}{2}} = \frac{\sqrt{5} + 1}{2}.$$

Therefore, the lemma holds. □

Theorem 3.8 *When all jobs have the same release date, Algorithm 2 is a 2-competitive on-line algorithm and the bound is tight.*

Proof For any $j \in \{1, 2, \dots, n\}$, let $A_{j_j}^*$ be the optimal solution for instance (J_j, π) , and let $Z_{j_j}^*$ be the optimal value. If $A_{j_j}^* = \emptyset$, since $A_{j_j,0} = \emptyset$ and $A_{j_j} = \arg \min_{A_{j_j,t}} Z_{j_j,t}$, we have $p(A_{j_j}) + \pi(\overline{A_{j_j}}) \leq Z_{j_j}^*$; otherwise, since all jobs have the same release date, we can resort the index of jobs to satisfy $j \in A_{j_j}^*$. Similar to inequality (4), we have $r_{t_{A_{j_j}}} + p(A_{j_j}) + \pi(\overline{A_{j_j}}) \leq Z_{j_j}^*$, where $t_{A_{j_j}} = \max\{j | j \in A_{j_j}\}$. Combining Lemma 3.4, the objective value of job set generated by Algorithm 2 is

$$C^j + \pi(\overline{A^j}) \leq Z_j^* + r_{t_{A_{j_j}}} + p(A_j) + \pi(\overline{A_{j_j}}) \leq 2Z_j^*.$$

To show that the bound is tight, we consider the following instance with n jobs, where $r_j = 0$ for $j \in \{1, 2, \dots, n\}$, $p_j = 1$ for $j \in \{1, 2, \dots, n - 1\}$, and $p_n = 3$. The submodular function $\pi(\cdot)$ is defined as follows:

$$\begin{cases} \pi(\emptyset) = 0; \\ \pi(\{n\}) = n; \\ \pi(S) = \sum_{j:j \in S} (1 + \frac{1}{n}), \text{ if } n \notin S; \\ \pi(S) = n + 1, \text{ otherwise.} \end{cases}$$

It can be verified that the optimal solution is to reject all jobs, and the optimal value is $Z^* = n + 1$. However, the output job set by Algorithm 2 is to contain the first n

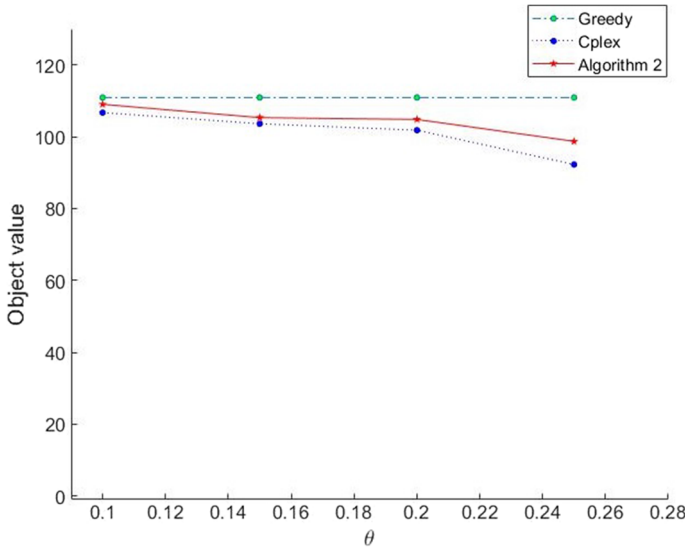


Fig. 1 Solution quality for n=10

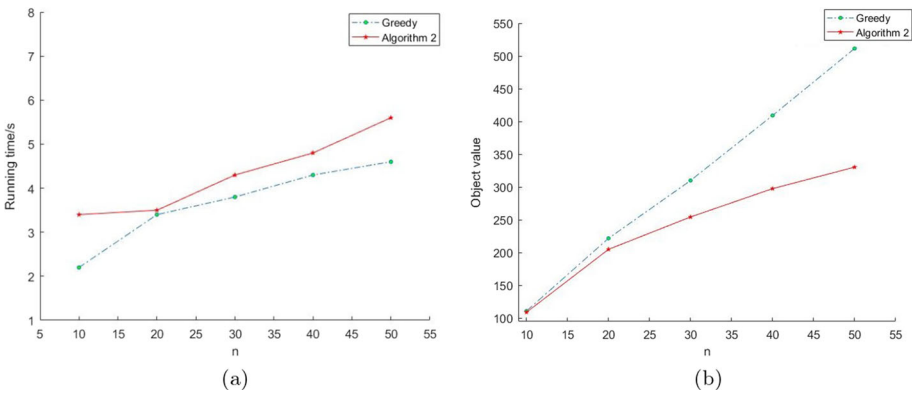


Fig. 2 Performances of Algorithm 2 against the greedy algorithm

jobs, and its objective value is $Z = 2n - 1$. Thus, we have

$$\frac{Z}{Z^*} = \frac{2n - 1}{n} \rightarrow 2, \text{ when } n \rightarrow \infty. \quad \square$$

4 Numerical experiments

The experimental study aims to present the practical performance of Algorithm 2, and compare against a baseline method.

Datasets. The processing time and the release date of any jobs in datasets are generated in an average distribution over a given range.

Submodular penalty function. The penalty function required by the experiment must be non-negative, monotonic, and submodular. We use the following submodular function for experiments,

$$\pi(S) = \sum_{J_j \in S} \pi(J_j) - \theta \cdot (|S|^2 - |S|), \quad \forall S (\neq \emptyset) \subseteq J,$$

where $\theta \geq 0$ is a parameter.

Experiment procedures. We use IBM's open-source tool Cplex to obtain the optimal solution for the single machine scheduling problem with release dates and submodular penalties. If we do not get the optimal solution within 5 min, we stop Cplex. For comparison, we also run the greedy algorithm (Lu et al. 2011) which is the on-line algorithm for the problem with linear penalties. The first experiment compares Algorithm 2 against the optimal solution generated by Cplex and the greedy algorithm in small datasets using different parameter θ . The second experiment compares Algorithm 2 against the greedy algorithm in the datasets with different number of jobs.

Implementation details. Implementation details. The whole experiment is implemented on a single process with an Intel(R) Core(TM) i5-9300H CPU at 2.40GHz and 8 GB RAM.

Numerical results. Figure 1 displays the objective values of Algorithm 2, the CPLEX and the greed algorithm in a small datasets. Figure2 displays the performances of Algorithm 2, and the greed algorithm in the datasets with different number of jobs. The results first show that the theoretical analysis indeed matches the practical performance. In addition, the gap between the objective values generated by Algorithm 2 and greedy algorithm is increasing as either θ or n increases, which is expected.

5 Conclusion

In this paper, we study the on-line single machine scheduling problem with release dates and submodular rejection penalties. We prove that there is no on-line algorithm with a constant competitive ratio if the penalty submodular function is not monotone, and present an on-line algorithm with a competitive ratio of 3 when the penalty submodular function is monotone. In particular, we consider the special case of this problem, where all jobs have the same release date. We prove that there is no on-line algorithm with a competitive ratio of $\frac{\sqrt{5}+1}{2} \approx 1.618$ even when the release date of each job is 0, and the competitive ratio of the on-line algorithm is 2.

It is challenging to either find a greater lower bound or design an on-line algorithm with a better competitive ratio. The information of jobs with the same release date can be revealed when these jobs arrive, and the on-line-over-time case of this problem is worth considering.

Acknowledgements The work is supported in part by the National Natural Science Foundation of China [grant No. 12071417], National Key R&D Program of China [grant No. 2020AAA 0105200], the Yunnan Provincial Research Foundation for Basic Research, China. [grant No. 202301AU070197], and the Project of Yunnan Provincial Department of Education Science Research Fund [grant No. 2023J0014].

Declarations

Competing Interest The authors declare that they have no known competing financial interests.

References

- Albers S (1999) Better bounds for online scheduling. *SIAM J Comput* 29(2):459–473
- Bartal Y, Leonardi S, Marchetti-Spaccamela A, Sgall J, Stougie L (2000) Multiprocessor scheduling with rejection. *SIAM J Discret Math* 13(1):64–78
- Fleischer R, Wahl M (2000) Online scheduling revisited. *J Sched* 3(6):343–353
- Graham R (1966) Bounds for certain multiprocessing anomalies. *Bell Syst Tech J* 45:1563–1581
- Graham R (1969) Bounds on multiprocessing timing anomalies. *SIAM J Appl Math* 17:416–429
- He C, Leung JYT, Lee K, Pinedo ML (2016) Improved algorithms for single machine scheduling with release dates and rejections. *OR* 14: 41–55
- Hochbaum D, Shmoys D (1987) Using dual approximation algorithms for scheduling problems: theoretical and practical results. *J ACM* 34:144–162
- Iwata S, Orlin JB (2009) A simple combinatorial algorithm for submodular function minimization. In: Mathieu C (ed) the twentieth annual ACM-SIAM symposium on discrete algorithms, SODA 2009. SIAM, USA, pp 1230–1237
- Jansen K, Porkolab L (2001) Improved approximation schemes for scheduling unrelated parallel machines. *Math Oper Res* 26(2):324–338
- Jansen K, Klein KM, Verschae J (2020) Closing the gap for makespan scheduling via sparsification techniques. *Math Oper Res* 45(4):1371–1392
- Lawler EL (1973) Optimal sequencing a single machine subject to precedence constraints. *Manage Sci*
- Li W, Li J, Zhang T (2012) Two approximation schemes for scheduling on parallel machines under a grade of service provision. *Asia-Pacific J Oper Res* 29(05):1250029
- Li W, Li J, Zhang X, Chen Z (2015) Penalty cost constrained identical parallel machine scheduling problem. *Theoret Comput Sci* 607:181–192
- Liu X, Li W (2020) Approximation algorithm for the single machine scheduling problem with release dates and submodular rejection penalty. *Mathematics* 8(1):133
- Liu X, Li W (2021) Approximation algorithms for the multiprocessor scheduling with submodular penalties. *Optim Lett* 15:2165–2180
- Lu L, Ng C, Zhang L (2011) Optimal algorithms for single-machine scheduling with rejection to minimize the makespan. *Int J Prod Econ* 130(2):153–158
- Ou J, Zhong X, Wang G (2015) An improved heuristic for parallel machine scheduling with rejection. *Eur J Oper Res* 241(3):653–661
- Ou J, Zhong X, Li CL (2016) Faster algorithms for single machine scheduling with release dates and rejection. *Inf Process Lett* 116:503–507
- Rudin JF (2001) Improved bounds for the on-line scheduling problem. PhD thesis, The University of Texas at Dallas
- Wang W, Liu X (2022) A Combinatorial 2-approximation algorithm for the parallel-machine scheduling with release times and submodular penalties. *Mathematics* 10:61
- Zhang L, Lu L, Yuan J (2009) Single machine scheduling with release dates and rejection. *Eur J Oper Res* 198(3):975–978
- Zhang X, Xu D, Du D, Wu C (2018) Approximation algorithms for precedence-constrained identical machine scheduling with rejection. *J Comb Optim* 35:318–330
- Zhang L, Lu L (2016) Parallel-machine scheduling with release dates and rejection. *4OR* 14, 165–172
- Zheng H, Gao S, Liu W, Wu W, Du D (2022) Approximation algorithm for the parallel-machine scheduling problem with release dates and submodular rejection penalties. *J Comb Optim* 44:343–353
- Zhong X, Pan Z, Jiang D (2017) Scheduling with release times and rejection on two parallel machines. *J Comb Optim* 33:934–944
- Zhong X, Ou J (2017) Improved approximation algorithms for parallel machine scheduling with release dates and job rejection. *4OR* 15, 387–406

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.