

Article

Approximation Algorithm for the Single Machine Scheduling Problem with Release Dates and Submodular Rejection Penalty

Xiaofei Liu ¹ and Weidong Li ^{2,*}¹ School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China; xiaofeil@pku.edu.cn² School of Mathematics and Statistics, Yunnan University, Kunming 650504, China

* Correspondence: weidong@ynu.edu.cn

Received: 11 November 2019; Accepted: 10 January 2020 ; Published: 16 January 2020



Abstract: In this paper, we consider the single machine scheduling problem with release dates and nonmonotone submodular rejection penalty. We are given a single machine and multiple jobs with probably different release dates and processing times. For each job, it is either accepted and processed on the machine or rejected. The objective is to minimize the sum of the makespan of the accepted jobs and the rejection penalty of the rejected jobs which is determined by a nonmonotone submodular function. We design a combinatorial algorithm based on the primal-dual framework to deal with the problem, and study its property under two cases. For the general case where the release dates can be different, the proposed algorithm have an approximation ratio of 2. When all the jobs release at the same time, the proposed algorithm becomes a polynomial-time exact algorithm.

Keywords: scheduling; submodular rejection penalty; approximation algorithm

1. Introduction

All jobs must be processed in classical scheduling problems. However, sometimes rejecting some jobs with low cost performance is possible and will bring more profit. This problem is called multiprocessor scheduling with rejection (MSR), which is first proposed by Bartal et al. [1]. In this problem, a job is either accepted and processed on the machine, or rejected and paid a rejection penalty. The objective is to minimize the makespan of accepted jobs plus the overall penalty of rejected jobs. Bartal et al. [1] proposed a 2-approximation algorithm running in $O(n \log n)$ time and a polynomial-time approximation scheme (PTAS) for MSR. Ou et al. [2] proposed a $(3/2 + \epsilon)$ -approximation algorithm with running time $O(n \log n + n/\epsilon)$ for MSR, where ϵ is a small given positive constant.

MSR and its variants have become increasingly popular in both academic and industrial community in the last two decades. Zhang and Lu [3] considered parallel-machine scheduling with release dates and rejection, where jobs cannot be processed before their corresponding release dates. The objective of this problem to minimize the sum of makespan of accepted jobs and total penalty of rejected jobs. They developed a 2-approximation algorithm. This result is further improved by Zhong and Ou [4] who designed a PTAS. Li et al. [5] designed a PTAS for a variant of MSR, where the objective is to minimize the makespan when the rejection cost is bounded by a given constant.

Some studies considered the case when the number of machines is bounded. When there are exactly two machines. Zhong et al. [6] developed a $(3/2 + \epsilon)$ -approximation algorithm for scheduling with release dates and rejection. When the number of machines is one, there are much more related results. Shabtay et al. [7] handled the single machine with job rejection and positional penalties problem

with a bicriteria approach. Zhang et al. [8] proved the NP-hardness of single machine scheduling with release dates and rejection, which is a special case of the problem considered in [3]. They presented a 2-approximation algorithm with running time of $O(n^2)$, which was independently improved by He et al. [9] and Ou et al. [10] with running time of $O(n \log n)$. Zhang et al. [11] developed a FPTAS for a variant of single machine scheduling with release dates and rejection, where the objective is to minimize the makespan when rejection penalty is bounded. Zou et al. [12] considered single-machine scheduling with rejection and an operator non-availability interval where jobs are prohibited to start or complete in the operator non-availability interval, and presented a FPTAS. Shioura et al. [13] studied a problem on a single machine with release dates and deadlines, where the processing can be accelerated by making additional cost. They utilized submodular optimization to achieve the best possible running time. Li and Cui [14] designed a 1.58-approximation algorithm for vector scheduling on a single machine, where vector scheduling imply that multiple resources are needed for each job and the objective is to minimize the maximal cost over all resources plus the rejection cost. More related results can be found in the surveys [15,16].

Since submodular function has the property of decreasing marginal return and occurs in many mathematical models, classical combinatorial optimization problems with submodular rejection penalty receive more and more attentions [17–19]. For the MSR problem, rejecting jobs frequently might lead to low prestige for the manufacturer. There is no linear relationship between the number of rejected jobs and the loss of the manufacturer's prestige, but as the number of rejected jobs increases, it will have less and less impact on the manufacturer's prestige. Recently, Zhang et al. [20] proposed a 3-approximation algorithm for precedence-constrained scheduling with submodular rejection on parallel machines.

Motivated by the problems in [8,20], we propose a new scheduling problem, called single machine scheduling problem with release dates and submodular rejection penalty, where each job is either rejected or accepted and processed on the machine. The objective is to minimize the sum of the makespan of the accepted jobs and the rejection penalty of the rejected jobs which is determined by a submodular function. In this paper, we design a combinatorial 2-approximation algorithm by using the primal-dual scheme, generalizing the algorithm presented in [8].

The remainder of this paper is structured as follows. In Section 2, we provide a formal problem statement. In Section 3, for single machine scheduling problem with release dates and submodular rejection penalty functions, we present a 2-approximation algorithm, based on a method similarly to the primal-dual algorithm. In addition, when all the jobs have the same release date, we present a polynomial-time exact algorithm. We provide conclusion in Section 4.

2. Preliminaries

In this section, we present a formal description of single machine scheduling problem with release dates and submodular rejection penalty. Given a single machine and n jobs, $J = \{J_1, J_2, \dots, J_n\}$, each job J_j has a processing time p_j and a release date r_j , where p_j and r_j are nonnegative real numbers. The rejection cost of a subset of jobs is determined by the nonmonotone submodular function $\pi(\cdot) : 2^J \rightarrow R^+ \cup \{0\}$, which means that

$$\pi(X \cup Y) + \pi(X \cap Y) \leq \pi(X) + \pi(Y), \forall X, Y \subseteq J,$$

without loss of generality, assume that $\pi(\emptyset) = 0$. Moreover, we assume that $\pi(S)$ can be computed within polynomial time for any subset $S \subseteq J$, where the 'polynomial' we use regard to the size n .

The single machine scheduling problem with release dates and submodular rejection penalties is to choose a set $R \subseteq J$ of rejected jobs and schedule the remaining jobs in $A = J \setminus R$ on the machine. The objective is to minimize the sum of the makespan of the accepted jobs and the penalty cost $\pi(R)$ of the rejected jobs. By using the general notation for scheduling problems, the problem is denoted

by $1|r_j, reject|C_{\max} + \pi(R)$. Clearly, if $\pi(R) = \sum_{J_j: J_j \in R} \pi(\{J_j\})$, the problem $1|r_j, reject|C_{\max} + \pi(R)$ is exactly the problem $1|r_j, reject|C_{\max} + \sum_{J_j \in R} w_j$ considered in [8].

If rejection is not allowed, Lawler [21] shows that the corresponding problem $1|r_j, reject|C_{\max}$ can be solved using the earliest release date rule (ERD-rule for short). Thus, we have the following lemma.

Lemma 1 ([21]). *For the problem $1|r_j, reject|C_{\max} + \pi(R)$, there exists an optimal schedule such that the accepted jobs in $A = J \setminus R$ are processed using the ERD-rule.*

For any set $S \subseteq J$, let

$$p(S) = \sum_{J_j \in S} p_j$$

be the total processing time of the jobs in S . Based on Lemma 1, we first re-label the jobs such that $r_1 \leq \dots \leq r_n$. For convenience, let

$$r_0 = 0.$$

Assume that $r_1 \geq 0$. For each $r \in \{r_0, r_1, r_2, \dots, r_n\}$, let

$$B(r) = \{J_j | r_j > r\}$$

be the set of jobs with release date bigger than r . Correspondingly, let B_r be the set of jobs such that

$$B_r \supseteq B(r), \text{ and } \pi(B_r) \text{ is minimized.}$$

Obviously, if $\pi(\cdot)$ is monotone nondecreasing, $B_r = B(r)$. Otherwise, we have

Lemma 2. *For every $r \in \{r_0, r_1, r_2, \dots, r_n\}$, B_r can be computed in polynomial time.*

Proof. Consider a given $r \in \{r_j | 0 \leq j \leq n\}$. Clearly, $\pi(B(r))$ is a constant. For any subset $X \subseteq J \setminus B(r)$, construct an auxiliary function $w(\cdot)$ such that

$$w(X) = \pi(X \cup B(r)) - \pi(B(r)).$$

For any two subsets $X, Y \subseteq J \setminus B(r)$, we have

$$\begin{aligned} & w(X) + w(Y) \\ &= \pi(X \cup B(r)) - \pi(B(r)) + \pi(Y \cup B(r)) - \pi(B(r)) \\ &\geq \pi((X \cup B(r)) \cup (Y \cup B(r))) + \pi((X \cup B(r)) \cap (Y \cup B(r))) - 2\pi(B(r)) \\ &= \pi((X \cup Y) \cup B(r)) - \pi(B(r)) + \pi((X \cap Y) \cup B(r)) - \pi(B(r)) \\ &= w(X \cup Y) + w(X \cap Y), \end{aligned}$$

where the inequality follows from that $\pi(\cdot)$ is a submodular function. Therefore, $w(\cdot)$ is a submodular function, which implies that $\min_{X \subseteq J \setminus B(r)} w(X)$ can be computed within polynomial time, using the method in [22]. Since $\pi(B(r))$ is a constant, $\min_{X \subseteq J \setminus B(r)} \pi(X \cup B(r)) = \min_{X \subseteq J \setminus B(r)} w(X) + \pi(B(r))$ can be computed within polynomial time. It implies that $B_r = (\arg \min_{X \subseteq J \setminus B(r)} \pi(X \cup B(r))) \cup B(r)$ can be computed within polynomial time. \square

3. An Approximation Algorithm

In this section, we design a combination 2-approximation algorithm for $1|r_j, reject|C_{\max} + \pi(R)$. In particular, when all the jobs have the same release dates, we prove that $1|r_j, reject|C_{\max} + \pi(R)$ can be solved exactly in polynomial time.

For each $r \in \{r_j | 0 \leq j \leq n\}$, use Process A(r), which is described later, to find a schedule σ_r , compute the objective value Z_r of schedule σ_r , and choose the best schedule as the output schedule.

In Process A(r), similarly to the primal-dual algorithm in [23], we introduce an auxiliary “dual” variable $\alpha_{r,j}$ for every job $J_j \in J$ and r , where r is an input value of Process A(r). There is an auxiliary notion of time t associated with Process A(r).

Process A(r).

Input: An instance of $1|r_j, reject|C_{\max} + \pi(R)$ and a fixed number $r \in \{r_j | 0 \leq j \leq n\}$.

Output: a schedule σ_r and its objective value Z_r .

Step 1. Initially, set the variable $\alpha_{r,j} = 0$ for each job J_j in J and the time $t = 0$.

Step 2. Compute the job set B_r using the method described in the proof of Lemma 2. Reject every job in B_r by setting $R_r := B_r$, and freeze every job in B_r by setting $F_r := B_r$.

Step 3. While $J \setminus F_r \neq \emptyset$, we increase the variables $\alpha_{r,j}$ for all unfrozen jobs in $J \setminus F_r$ uniformly at unit rate with time t . As t increases, one of the following two cases may occur:

Case 1. There is a set $S_r \subseteq J$ and $S_r \cap F_r \neq \emptyset$ satisfying that

$$\sum_{j: J_j \in S_r \setminus F_r} t + \sum_{j: J_j \in S_r \cap F_r} \alpha_{r,j} + \pi(B_r) = \pi(S_r \cup B_r). \quad (1)$$

In this case, set $\alpha_{r,j} = t$ for each job J_j in $S_r \setminus F_r$ and freeze those unfrozen jobs in S_r , by setting $F_r := F_r \cup S_r$. Reject all jobs in set S_r , by setting $R_r := R_r \cup S_r$.

Case 2. There is a job $J_j \in J \setminus F_r$ such that $t = p_j$. In this case, set $\alpha_{r,j} = t$ and freeze job J_j by setting $F_r := F_r \cup \{J_j\}$.

Step 4. Reject the jobs in R_r and schedule all jobs in $A_r (= J \setminus R_r)$ in time interval $[r, r + p(A_r)]$ on the machine. The resulting schedule is denoted by σ_r , whose objective value is denoted by Z_r .

In Process A(r) for any $r = \{r_1, \dots, r_n\}$ in the algorithm, at Step 2, we can find the exact set B_r for a given value r , and B_r will not change until Process A(r) terminates. However, at Step 3 of Process A(r) for any $r = \{r_1, \dots, r_n\}$ in the algorithm, the frozen job set F_r and job set S_r are changed as the change as time t . And the auxiliary “dual” variable $\alpha_{r,j}$ for any $J_j \in J$ and a given value r will increase with time t until job J_j is frozen. Thus, at any time t in Process A(r) for any $r = \{r_1, \dots, r_n\}$, we introduce the following notations:

- B_r is the set of jobs generated by Step 2 in ProcessA(r),
- $F_r(t)$ is the set of frozen jobs at time t in ProcessA(r),
- $S_r(t)$ is the set of jobs generated by Case 1 at time t in ProcessA(r).
- $\alpha_{r,j}(t)$ is the value of job J_j at time t in ProcessA(r),
- $\alpha_{r,j}$ is the value of job J_j when job J_j is frozen in ProcessA(r).

Lemma 3. For any $r \in \{r_0, r_1, \dots, r_n\}$, Process A(r) can be implemented in polynomial time.

Proof. Consider Step 2 of Process A(r), by Lemma 2, B_r can be found in polynomial time. Clearly, $\pi(B_r)$ is a constant.

Consider Step 3 of Process A(r). For any time \tilde{t} , let $F_r(\tilde{t})$ be the frozen job set at this time and \bar{t} be the next closest time, which is incurred by either Case 1 or Case 2.

For Case 1, the next closest time \tilde{t} is at most

$$\begin{aligned}\Delta_1 &= \min_{S \subseteq J, S \cap F_r(\tilde{t}) \neq \emptyset} \frac{\pi(S \cup B_r) - \pi(B_r) - \sum_{j: J_j \in S \cap F_r(\tilde{t})} \alpha_{r,j}}{|S \setminus F_r(\tilde{t})|} \\ &= \min_{S \subseteq J, S \cap F_r(\tilde{t}) \neq \emptyset} \frac{w(S) + \alpha(S)}{k(S)},\end{aligned}$$

where $w(S) = \pi(S \cup B_r) - \pi(B_r)$, $\alpha(S) = \sum_{j: J_j \in S \cap F_r(\tilde{t})} (-\alpha_{r,j})$ and $k(S) = |S \setminus F_r(\tilde{t})|$ for any subset $S \cap F_r(\tilde{t}) \neq \emptyset$. For any two subsets $S_1, S_2 \subseteq J$ satisfying $S_1 \cap F_r(\tilde{t}) \neq \emptyset$ and $S_2 \cap F_r(\tilde{t}) \neq \emptyset$, we have

$$\begin{aligned}&\alpha(S_1) + \alpha(S_2) \\ &= \sum_{j: J_j \in S_1 \cap F_r(\tilde{t})} (-\alpha_{r,j}) + \sum_{j: J_j \in S_2 \cap F_r(\tilde{t})} (-\alpha_{r,j}) \\ &= \sum_{j: J_j \in (S_1 \cap F_r(\tilde{t})) \cup (S_2 \cap F_r(\tilde{t}))} (-\alpha_{r,j}) + \sum_{j: J_j \in (S_1 \cap F_r(\tilde{t})) \cap (S_2 \cap F_r(\tilde{t}))} (-\alpha_{r,j}) \\ &= \sum_{j: J_j \in (S_1 \cup S_2) \cap F_r(\tilde{t})} (-\alpha_{r,j}) + \sum_{j: J_j \in (S_1 \cap S_2) \cap F_r(\tilde{t})} (-\alpha_{r,j}) \\ &= \alpha(S_1 \cup S_2) + \alpha(S_1 \cap S_2),\end{aligned}$$

and

$$\begin{aligned}&k(S_1) + k(S_2) \\ &= |S_1 \setminus F_r(\tilde{t})| + |S_2 \setminus F_r(\tilde{t})| \\ &= |(S_1 \setminus F_r(\tilde{t})) \cup (S_2 \setminus F_r(\tilde{t}))| + |(S_1 \setminus F_r(\tilde{t})) \cap (S_2 \setminus F_r(\tilde{t}))| \\ &= |(S_1 \cup S_2) \setminus F_r(\tilde{t})| + |(S_1 \cap S_2) \setminus F_r(\tilde{t})| \\ &= k(S_1 \cup S_2) + k(S_1 \cap S_2).\end{aligned}$$

Therefore, $\alpha(\cdot)$ and $k(\cdot)$ are modular functions. Similarly to the proof of Lemma 2, we can prove that $w(\cdot)$ is a submodular function. Therefore, $w(S) + \alpha(S)$ is a submodular function, which implies that the value of Δ_1 can be computed in polynomial time by using the combinatorial algorithm for the ratio of a submodular and a modular function minimization problem [24].

For Case 2, the next closet time \tilde{t} is at most

$$\Delta_2 = \min_{j: J_j \in J \setminus F_r(\tilde{t})} p_j.$$

It is obvious that the value of Δ_2 can be found in polynomial time.

Thus, the next time $\tilde{t} = \min\{\Delta_1, \Delta_2\}$ can be found in polynomial time. When the time increase from \tilde{t} to \tilde{t} , the algorithm freezes at least one unfrozen job, which implies that Step 3 of Process A(r) can be completed in polynomial time. It is easy to verify other Steps can be implemented in polynomial time. Thus, the lemma holds. \square

Lemma 4. For any $r \in \{r_j | 0 \leq j \leq n\}$, the rejected job set R_r produced by Process A(r) satisfies

$$\pi(R_r) = \sum_{j: J_j \in R_r} \alpha_{r,j} + \pi(B_r). \quad (2)$$

Proof. For any $r \in \{r_j | 0 \leq j \leq n\}$, let $\alpha_{r,j}(t)$ be the variable of job J_j at time t which will increase with time t until job J_j is frozen. Thus, if J_j is frozen at time t' , we have

$$\begin{cases} \alpha_{r,j}(t_1) < \alpha_{r,j}(t_2) \text{ for } t_1 < t_2 \leq t'; \\ \alpha_{r,j}(t_1) = \alpha_{r,j}(t_2) \text{ for } t' \leq t_1 < t_2; \\ \alpha_{r,j} = \alpha_{r,j}(t') = t'. \end{cases} \quad (3)$$

By Step 2 of Process $A(r)$, jobs in set B_r are found and are frozen, we have $\pi(B_r)$ is a constant and

$$\alpha_{r,j} = 0 \text{ for } \forall J_j \in B_r.$$

By Case 1 of Process $A(r)$, for any time t and any subset $S \subseteq J$, we have

$$\sum_{j:J_j \in S} \alpha_{r,j}(t) + \pi(B_r) = \sum_{j:J_j \in S \setminus F_r(t)} t + \sum_{j:J_j \in S \cap F_r(t)} \alpha_{r,j} + \pi(B_r) \leq \pi(S \cup B_r), \quad (4)$$

where $F_r(t)$ is the frozen jobs set at time t . At time points t_1 and t_2 ($t_1 \leq t_2$) in Case 1, let $S_r(t_1)$ and $S_r(t_2)$ be the selected job sets, respectively. Thus, both $S_r(t_1)$ and $S_r(t_2)$ satisfy inequality (1), i.e., we have

$$\begin{aligned} \pi(S_r(t_1) \cup B_r) &= \sum_{j:J_j \in S_r(t_1) \setminus F_r(t_1)} t_1 + \sum_{j:J_j \in S_r(t_1) \cap F_r(t_1)} \alpha_{r,j} + \pi(B_r) = \sum_{j:J_j \in S_r(t_1)} \alpha_{r,j}(t_1) + \pi(B_r); \\ \pi(S_r(t_2) \cup B_r) &= \sum_{j:J_j \in S_r(t_2) \setminus F_r(t_2)} t_2 + \sum_{j:J_j \in S_r(t_2) \cap F_r(t_2)} \alpha_{r,j} + \pi(B_r) = \sum_{j:J_j \in S_r(t_2)} \alpha_{r,j}(t_2) + \pi(B_r), \end{aligned}$$

where all jobs in $S_r(t_1)$ (or $S_r(t_2)$) are frozen at time t_1 (or t_2) or even earlier.

Therefore, we have

$$\begin{aligned} &\sum_{j:J_j \in S_r(t_1) \cup S_r(t_2)} \alpha_{r,j}(t_2) + \pi(B_r) + \sum_{j:J_j \in S_r(t_1) \cap S_r(t_2)} \alpha_{r,j}(t_2) + \pi(B_r) \\ &= \sum_{j:J_j \in S_r(t_1)} \alpha_{r,j}(t_2) + \pi(B_r) + \sum_{j:J_j \in S_r(t_2)} \alpha_{r,j}(t_2) + \pi(B_r) \\ &\geq \sum_{j:J_j \in S_r(t_1)} \alpha_{r,j}(t_1) + \pi(B_r) + \sum_{j:J_j \in S_r(t_2)} \alpha_{r,j}(t_2) + \pi(B_r) \\ &= \pi(S_r(t_1) \cup B_r) + \pi(S_r(t_2) \cup B_r) \\ &\geq \pi((S_r(t_1) \cup S_r(t_2)) \cup B_r) + \pi((S_r(t_1) \cap S_r(t_2)) \cup B_r) \\ &\geq \pi((S_r(t_1) \cup S_r(t_2)) \cup B_r) + \sum_{j:J_j \in S_r(t_1) \cap S_r(t_2)} \alpha_{r,j}(t_2) + \pi(B_r), \end{aligned}$$

where the first inequality follows from $t_1 \leq t_2$, the second inequality follows from the submodularity of $\pi(\cdot)$, and the third inequality follows from inequality (4).

It implies that

$$\sum_{j:J_j \in S_r(t_1) \cup S_r(t_2)} \alpha_{r,j}(t_2) + \pi(B_r) \geq \pi((S_r(t_1) \cup S_r(t_2)) \cup B_r).$$

Moreover, $\sum_{j:J_j \in S_r(t_1) \cup S_r(t_2)} \alpha_{r,j}(t_2) + \pi(B_r) = \pi((S_r(t_1) \cup S_r(t_2)) \cup B_r)$ following from inequality (4). Any job $J_j \in S_r(t_1) \cup S_r(t_2)$ is frozen at t_2 or even earlier, we have

$$\pi((S_r(t_1) \cup S_r(t_2)) \cup B_r) = \sum_{j:J_j \in S_r(t_1) \cup S_r(t_2)} \alpha_{r,j}(t_2) + \pi(B_r) = \sum_{j:J_j \in S_r(t_1) \cup S_r(t_2)} \alpha_{r,j} + \pi(B_r).$$

By Process A(r), $R_r = B_r \cup \bigcup_{S \in \mathcal{S}} S$, where \mathcal{S} is the set of subset satisfying inequality (1), so the equality (2) holds. \square

Consider the optimal schedule σ^* . Let A^* and R^* be the sets of the accepted jobs and the rejected jobs in σ^* , respectively. Let $r^* = \max\{r_j | J_j \in A^*\}$ be the maximum release date of the jobs in A^* , where $r^* = 0$ if $A^* = \emptyset$.

Lemma 5. *There is an optimal schedule σ^* satisfying $B_{r^*} \subseteq R^*$.*

Proof. For any optimal schedule σ^* , If $B_{r^*} \setminus R^* = \emptyset$, the lemma holds. Otherwise, let σ' be a schedule that the jobs in $R' = R^* \cup B_{r^*}$ are rejected and the jobs in $A' = J \setminus (R^* \cup B_{r^*})$ are accepted and processed in ERD-rule. Clearly, we have $A' \subset A^* = J \setminus R^*$, which implies that σ' is an optimal schedule to process the jobs in A' by Lemma 1. Thus, the makespan of σ' is no more than that of σ^* .

By the definition of $B(r^*)$, we have $B(r^*) = \{J_j \in J | r_j > r^*\} \subseteq R^*$. Since $B(r^*) \subseteq B_{r^*}$, we have

$$B(r^*) \subseteq R^* \cap B_{r^*}.$$

By the definition of B_{r^*} , we have $\pi(B_{r^*}) \leq \pi(S)$ for any $S \supseteq B(r^*)$. That implies

$$\pi(B_{r^*}) \leq \pi(R^* \cap B_{r^*}).$$

Since function $\pi(\cdot)$ is a submodular function, we have

$$\pi(R^*) + \pi(B_{r^*}) \geq \pi(R^* \cup B_{r^*}) + \pi(R^* \cap B_{r^*}) = \pi(R') + \pi(R^* \cap B_{r^*}).$$

Together with $\pi(B_{r^*}) \leq \pi(R^* \cap B_{r^*})$, we have

$$\pi(R') \leq \pi(R^*).$$

Therefore, we have that the objective value of σ' is no more than that of σ^* . Thus, σ' is an optimal scheduling that $R' = R^* \cup B_{r^*}$ is the rejected set of σ' . The lemma holds. \square

Let σ be the output schedule. Let Z and Z^* be the objective values of the schedule σ and the optimal schedule σ^* , respectively.

Theorem 1. $Z \leq 2Z^*$ and the bound is tight.

Proof. Let σ^* be an optimal schedule such that $B_{r^*} \subseteq R^*$. If the optimal schedule σ^* rejects all the jobs, then σ^* is the output schedule of Process A(r_0) where $r_0 = 0$, which implies that the theorem holds. Otherwise, let $r^* = \max\{r_j | J_j \in A^*\}$ be the maximum release date of the jobs in A^* , we have

$$Z^* \geq r^*. \quad (5)$$

By Case 2 of Process A(r^*), we have

$$\begin{cases} \alpha_{r^*,j} \leq p_j \text{ for any job } J_j \in J; \\ \alpha_{r^*,j} = p_j \text{ for any job } J_j \in A_{r^*}, \end{cases} \quad (6)$$

where $\alpha_{r^*,j}$ is the frozen variable of job J_j and A_{r^*} is the accepted jobs produced by Process A(r^*). Since A^* and R^* are the sets of the accepted jobs and the rejected jobs in optimal schedule σ^* , we have

$$\begin{aligned}
 Z^* &\geq p(A^*) + \pi(R^*) \\
 &= \sum_{j:J_j \in A^*} p_j + \pi(R^* \cup B_{r^*}) \\
 &\geq \sum_{j:J_j \in A^*} p_j + \sum_{j:J_j \in R^*} \alpha_{r^*,j} + \pi(B_{r^*}) \\
 &\geq \sum_{j:J_j \in J} \alpha_{r^*,j} + \pi(B_{r^*}), \tag{7}
 \end{aligned}$$

where the second inequality follows from inequality (4) and the third inequality follows from inequality (6).

Therefore, we have

$$\begin{aligned}
 Z &\leq Z_{r^*} = r^* + p(A_{r^*}) + \pi(R_{r^*}) \\
 &= r^* + \sum_{j:J_j \in A_{r^*}} \alpha_{r^*,j} + \sum_{j:J_j \in R_{r^*}} \alpha_{r^*,j} + \pi(B_{r^*}) \\
 &= r^* + \sum_{j:J_j \in J} \alpha_{r^*,j} + \pi(B_{r^*}) \\
 &\leq 2Z^*,
 \end{aligned}$$

where the second equality follows equality (6) and Lemma 4, and the last inequality follows from inequalities (5) and (7).

To show that the bound is tight, consider the following instance with three jobs: $J_1 = (r_1, p_1) = (1, 2)$, $J_2 = (r_2, p_2) = (1, 1)$ and $J_3 = (r_3, p_3) = (4, 0)$. The submodular function $\pi(\cdot)$ is defined as following: $\pi(\{J_1\}) = 3$, $\pi(\{J_2\}) = 1$, $\pi(\{J_3\}) = 10$, $\pi(\{J_1, J_2\}) = 4$, $\pi(\{J_1, J_3\}) = 12$, $\pi(\{J_2, J_3\}) = 11$ and $\pi(\{J_1, J_2, J_3\}) = 13$.

Through the Process A(r), when $r = r_0 = 0$, the resulting schedule σ_{r_0} is to reject J_1, J_2, J_3 . Thus, we have $Z_{r_0} = \pi(\{J_1, J_2, J_3\}) = 13$; when $r = r_1$ (or r_2) = 1, the resulting schedule σ_{r_1} (or σ_{r_2}) is to accept J_2 and reject J_1, J_3 . Thus, we have $Z_{r_1} (= Z_{r_2}) = r_1 + p_2 + \pi(\{J_1, J_3\}) = 14$; when $r = r_3 = 4$, the resulting schedule σ_{r_3} is to accept J_1, J_3 and reject J_2 . Thus, we have $Z_3 = r_3 + p_1 + p_3 + \pi(\{J_2\}) = 8$. The optimal schedule is to accept J_1, J_2, J_3 . And we have $Z^* = r_1 + p_1 + p_2 + p_3 = 4$. Thus, we have $Z = \min\{Z_{r_0}, Z_{r_1}, Z_{r_2}, Z_{r_3}\} = 8 = 2Z^*$. \square

Then, we consider a special case for the problem $1|r_j, reject|C_{\max} + \pi(R)$ where all the jobs have the same release date.

Theorem 2. *The problem $1|r_j, reject|C_{\max} + \pi(R)$ can be solved optimally in polynomial time when all the jobs have the same release date.*

Proof. If the optimal schedule σ^* rejected all the jobs, then σ^* is the output schedule of Process A(r_0), where $r_0 = 0$, the theorem holds. Otherwise, assume $r_j = r^*$ for any job $J_j \in J$. The start processing time of machine is r^* and there is no job J_j in J satisfying $r_j > r^*$. That implies that

$$B_{r^*} = \emptyset \text{ and } \pi(B_{r^*}) = 0.$$

Thus, we have

$$\begin{aligned}
 Z^* &= r^* + p(A^*) + \pi(R^*) \\
 &= r^* + \sum_{j:J_j \in A^*} p_j + \pi(R^* \cup B_{r^*}) \\
 &\geq r^* + \sum_{j:J_j \in A^*} \alpha_{r^*,j} + \sum_{j:J_j \in R^*} \alpha_{r^*,j} + \pi(B_{r^*}) \\
 &= r^* + \sum_{j:J_j \in I} \alpha_{r^*,j},
 \end{aligned} \tag{8}$$

where the inequality follows from inequalities (4) and (6).

Through Step 3 of Process A(r), we have

$$\begin{aligned}
 Z &\leq Z_{r^*} = r^* + p(A_{r^*}) + \pi(R_{r^*}) \\
 &= r^* + \sum_{j:J_j \in A_{r^*}} \alpha_{r^*,j} + \sum_{j:J_j \in R_{r^*}} \alpha_{r^*,j} + \pi(B_{r^*}) \\
 &= r^* + \sum_{j:J_j \in I} \alpha_{r^*,j} \\
 &\leq Z^*,
 \end{aligned}$$

where the second equality follows equality (6) and Lemma 4, and the last equality follows from $\pi(B_{r^*}) = 0$.

Therefore, $Z = Z^*$, implying that the theorem holds. \square

4. Conclusions and Future Work

In this paper, we proposed the single machine scheduling problem $1|r_j, reject|C_{\max} + \pi(R)$ with release dates and nonmonotone submodular penalty, which generalizes the single machine scheduling with release dates and rejection [8]. We designed a combinatorial 2-approximation algorithm running in polynomial time with respect to this problem, base on the primal-dual framework. Noting that a combinatorial algorithm [8] only achieved an approximation ratio of 2 when the penalty function is modular, we extend the modular penalty function into submodular one without increasing the approximation ratio. In addition, when all the jobs have the same release date, exactly the same algorithm will become an exact algorithm running in polynomial time.

The topic could be further studied in the following ways. The online version of this problem is worth considering, since we do not always know complete information about jobs that have not arrived in real world. It is also challenging to design a fully polynomial-time approximation scheme for $1|r_j, reject|C_{\max} + \pi(R)$ as in [8]. Moreover, when there are multiple machines rather than a single machine, algorithms under this setting could be further developed.

Author Contributions: Conceptualization, X.L. and W.L.; methodology, W.L.; software, X.L.; validation, X.L. and W.L.; formal analysis, X.L.; investigation, W.L.; resources, W.L.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, W.L.; visualization, X.L.; supervision, W.L.; project administration, W.L.; funding acquisition, W.L. All authors have read and agreed to the published version of the manuscript.

Funding: The work is supported in part by the National Natural Science Foundation of China [No. 61662088], Program for Excellent Young Talents of Yunnan University, Training Program of National Science Fund for Distinguished Young Scholars, Project for Innovation Team (Cultivation) of Yunnan Province, IRTSTYN, and Key Joint Project of the Science and Technology Department of Yunnan Province and Yunnan University [No. 2018FY001(-014)].

Conflicts of Interest: The authors declare no conflict of interest

References

1. Bartal, Y.; Leonardi, S.; Marchetti-Spaccamela, A.; Sgall, J.S.L. Multiprocessor scheduling with rejection. *SIAM J. Discret. Math.* **2000**, *13*, 64–78. [\[CrossRef\]](#)
2. Ou, J.; Zhong, X.; Wang, G. An improved heuristic for parallel machine scheduling with rejection. *Eur. J. Oper. Res.* **2015**, *241*, 653–661. [\[CrossRef\]](#)
3. Zhang, L.; Lu, L. Parallel-machine scheduling with release dates and rejection. *4OR Q. J. Oper. Res.* **2016**, *14*, 165–172. [\[CrossRef\]](#)
4. Zhong, X.; Ou, J. Improved approximation algorithms for parallel machine scheduling with release dates and job rejection. *4OR Q. J. Oper. Res.* **2017**, *15*, 387–406. [\[CrossRef\]](#)
5. Li, W.; Li, J.; Zhang, X.; Chen, Z. Penalty cost constrained identical parallel machine scheduling problem. *Theor. Comput. Sci.* **2015**, *607*, 181–192. [\[CrossRef\]](#)
6. Zhong, X.; Pan, Z.; Jiang, D. Scheduling with release times and rejection on two parallel machines. *J. Comb. Optim.* **2017**, *33*, 934–944. [\[CrossRef\]](#)
7. Shabtay, D.; Gaspar, N.; Yedidsion, L. A bicriteria approach to scheduling a single machine with job rejection and positional penalties. *J. Comb. Optim.* **2012**, *23*, 395–424. [\[CrossRef\]](#)
8. Zhang, L.; Lu, L.; Yuan, J. Single machine scheduling with release dates and rejection. *Eur. J. Oper. Res.* **2009**, *198*, 975–978. [\[CrossRef\]](#)
9. He, C.; Leung, J.Y.T.; Lee, K.; Pinedo, M.L. Improved algorithms for single machine scheduling with release dates and rejections. *4OR Q. J. Oper. Res.* **2016**, *14*, 41–55. [\[CrossRef\]](#)
10. Ou, J.; Zhong, X.; Li, C.L. Faster algorithms for single machine scheduling with release dates and rejection. *Inform. Process. Lett.* **2016**, *116*, 503–507. [\[CrossRef\]](#)
11. Zhang, L.; Lu, L.; Yuan, J. Single-machine scheduling under the job rejection constraint. *Theor. Comput. Sci.* **2010**, *411*, 1877–1882. [\[CrossRef\]](#)
12. Zou, L.; Sun, Z.; Lu, L.; Zhang, L. Single-machine scheduling with rejection and an operator non-availability interval. *Mathematics* **2019**, *7*, 668.
13. Shioura, A.; Shakhlevich, N.V.; Strusevich, V.A. Application of submodular optimization to single machine scheduling with controllable processing times subject to release dates and deadlines. *INFORMS J. Comput.* **2016**, *28*, 148–161. [\[CrossRef\]](#)
14. Li, W.; Cui, Q. Vector scheduling with rejection on a single machine. *4OR Q. J. Oper. Res.* **2018**, *16*, 95–104. [\[CrossRef\]](#)
15. Shabtay, D.; Gaspar, N.; Kaspi, M. A survey on offline scheduling with rejection. *J. Sched.* **2013**, *16*, 3–28. [\[CrossRef\]](#)
16. Sotskov, Y.N.; Egorova, N.G. The optimality region for a single-machine scheduling problem with bounded durations of the jobs and the total completion time objective. *Mathematics* **2019**, *7*, 382. [\[CrossRef\]](#)
17. Du, D.; Lu, R.; Xu, D. A primal-dual approximation algorithm for the facility location problem with submodular penalties. *Algorithmica* **2012**, *63*, 191–200. [\[CrossRef\]](#)
18. Sharma, Y.; Swamy, C.; Williamson, D.P. Approximation algorithms for prize collecting forest problems with submodular penalty functions. In Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007; pp. 1275–1284.
19. Li, Y.; Du, D.; Xiu, N.; Xu, D. Improved approximation algorithms for the facility location problems with linear/submodular penalties. *Algorithmica* **2015**, *73*, 460–482. [\[CrossRef\]](#)
20. Zhang, X.; Xu, D.; Du, D.; Wu, C. Approximation algorithms for precedence-constrained identical machine scheduling with rejection. *J. Comb. Optim.* **2018**, *35*, 318–330. [\[CrossRef\]](#)
21. Lawler, E.L. Optimal sequencing a single machine subject to precedence constraints. *Manag. Sci.* **1973**, *19*, 544–546. [\[CrossRef\]](#)
22. Iwata, S.; Fleischer, L.; Fujishige, S. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *J. ACM* **2001**, *48*, 761–777. [\[CrossRef\]](#)

23. Xu, D.; Wang, F.; Du, D.; Wu, C. Approximation algorithms for submodular vertex cover problems with linear/submodular penalties using primal-dual technique. *Theor. Comput. Sci.* **2016**, *630*, 117–125. [[CrossRef](#)]
24. Fleischer, L.; Iwata, S. A push-relabel framework for submodular function minimization and applications to parametric optimization. *Discret. Appl. Math.* **2003**, *131*, 311–322. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).