

Article

A Combinatorial 2-Approximation Algorithm for the Parallel-Machine Scheduling with Release Times and Submodular Penalties

Wencheng Wang ^{1,*} and Xiaofei Liu ^{2,3,*}¹ School of Mathematics and Statistics, Yunnan University, Kunming 650500, China² School of Information Science and Engineering, Yunnan University, Kunming 650500, China³ School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China

* Correspondence: Wencheng@ynu.edu.cn (W.W.); xiaofei@ynu.edu.cn (X.L.)

† These authors contributed equally to this work.

Abstract: In this paper, we consider parallel-machine scheduling with release times and submodular penalties ($P|r_j, reject|C_{\max} + \pi(R)$), in which each job can be accepted and processed on one of m identical parallel machines or rejected, but a penalty must be paid if a job is rejected. Each job has a release time and a processing time, and the job can not be processed before its release time. The objective of $P|r_j, reject|C_{\max} + \pi(R)$ is to minimize the makespan of the accepted jobs plus the penalty of the rejected jobs, where the penalty is determined by a submodular function. This problem generalizes a multiprocessor scheduling problem with rejection, the parallel-machine scheduling with submodular penalties, and the single machine scheduling problem with release dates and submodular rejection penalties. In this paper, inspired by the primal-dual method, we present a combinatorial 2-approximation algorithm to $P|r_j, reject|C_{\max} + \pi(R)$. This ratio coincides with the best known ratio for the parallel-machine scheduling with submodular penalties and the single machine scheduling problem with release dates and submodular rejection penalties.



Citation: Wang, W.; Liu, X. A Combinatorial 2-Approximation Algorithm for the Parallel-Machine Scheduling with Release Times and Submodular Penalties. *Mathematics* **2022**, *10*, 61. <https://doi.org/10.3390/math10010061>

Academic Editor: Frank Werner

Received: 16 November 2021

Accepted: 13 December 2021

Published: 25 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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 (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: parallel-machine scheduling; submodular rejection penalty; approximation algorithm

1. Introduction

All jobs must be accepted and processed in classical scheduling problems [1–4]. However, to gain more profit, we can reject some jobs that have a larger processing time and result in smaller profits. Bartal et al. [5] first addressed the multiprocessor scheduling problem with rejection (MSR), in which the jobs can be rejected and a penalty must be paid for each rejected job. The objective is to minimize the makespan of the accepted jobs plus the total penalty of the rejected jobs. For the MSR, Bartal et al. [5] proposed a 2-approximation algorithm in time $O(n \log n)$ and a polynomial-time approximation scheme (PTAS). Later, Ou et al. [6] improved a $(3/2+\epsilon)$ -approximation algorithm in time $O(n \log n + \frac{n}{\epsilon})$, where $\epsilon > 0$ can be any small given constant.

Variants of the MSR have been studied extensively [7–9]. Zhang et al. [7] considered single machine scheduling with release dates and rejection, where jobs cannot be processed before their corresponding release dates. The objective is to minimize the makespan of the accepted jobs plus the total penalty of the rejected jobs. They proved that this problem is NP-hard, and presented a 2-approximation algorithm and a fully polynomial-time approximation scheme (FPTAS). Zhong et al. [10] considered two parallel-machine scheduling with release dates and rejection, and presented a $(3/2+\epsilon)$ -approximation algorithm with time complexity $O((\frac{n}{\epsilon})^2)$, where ϵ is any given small positive constant. Zhang and Lu [8] considered parallel-machine scheduling with release dates and rejection, and presented a 2-approximation algorithm. In particular, when m is a fixed constant, Zhang and Lu [8] designed an FPTAS.

A set function $f(\cdot)$ of J is a mapping from all subsets of J to real numbers, i.e., $f(\cdot): 2^J \rightarrow \mathbb{R}$. A set function $f(\cdot)$ is submodular if it satisfies $f(X \cup Y) + f(X \cap Y) \leq f(X) + f(Y)$, $\forall X, Y \subseteq J$, which has the property of decreasing marginal return. Recently, submodular functions have played a key role in the field of combinatorial optimization [9,11–13]. Liu and Li [14] considered parallel-machine scheduling with submodular penalties and proposed a $(2 - \frac{1}{m})$ -approximation algorithm based on the greedy method and list scheduling algorithm. Zhang et al. [15] considered precedence-constrained scheduling with submodular rejection on parallel machines, and proposed a 3-approximation algorithms. Based on the primal-dual method, Liu and Li presented a 2-approximation algorithm for [16] single machine scheduling with release dates and submodular rejection penalty. More related results can be found in the surveys [17–23].

Motivated by the optimization problems mentioned-above, we consider parallel-machine scheduling with release times and submodular penalties ($P|r_j, reject|C_{\max} + \pi(R)$), which is defined as follows.

Given a set $J = \{J_1, J_2, \dots, J_n\}$ of n jobs and a set $M = \{M_1, M_2, \dots, M_m\}$ of m parallel machines, each job $J_j \in J$ has a processing time $p_j (\geq 0)$ and a release time $r_j (\geq 0)$, where the job can be processed at or after its release time, without loss of generality, we assume that $\min_{j: J_j \in J} r_j = 0$. For the penalty submodular function $\pi(\cdot): 2^J \rightarrow \mathbb{R}_{\geq 0}$, without loss of generality, we assume that $\pi(\emptyset) = 0$. The $P|r_j, reject|C_{\max} + \pi(R)$ is to find a rejected set R . The objective is to minimize the makespan of the accepted jobs $J \setminus R$ plus the penalty of R , where the penalty is determined by penalty submodular function $\pi(\cdot)$.

Clearly, if $r_j = 0$ for $J_j \in J$, the $P|r_j, reject|C_{\max} + \pi(R)$ problem is exactly the parallel-machine scheduling with submodular penalties considered in [14]; If the rejection cost function is linear, the $P|r_j, reject|C_{\max} + \pi(R)$ problem is exactly the parallel-machine scheduling with penalties considered in [9]; If $m = 1$, the $P|r_j, reject|C_{\max} + \pi(R)$ problem is exactly the single machine scheduling problem with release dates and submodular rejection penalty considered in [16].

A difficulty of implementing the algorithm presented in [9] on the $P|r_j, reject|C_{\max} + \pi(R)$ problem is that the release time of the jobs is different and the jobs cannot be processed immediately in the given order. In order to overcome this problem, using the traversal method, we determine the set of jobs with designated release time and unify the releasing time of the other jobs. Then, in this paper, we present a combinatorial 2-approximation algorithm for $P|r_j, reject|C_{\max} + \pi(R)$. This ratio coincides with the best known ratio for the parallel-machine scheduling with submodular penalties and the single machine scheduling problem with release dates and submodular rejection penalties.

The structure of this paper is organized as follows. In Section 2, we present some terminologies and fundamental lemmas. In Section 3, we provide the 2-approximation algorithm for the $P|r_j, reject|C_{\max} + \pi(R)$. In Section 4, we present our conclusions.

2. Terminologies and Key Lemmas

Zhang and Lu [8] showed that the $P|r_j|C_{\max}$ problem can be solved by the earliest release date (ERD)-rule. That is, whenever some machine is idle and some job is available, process the unscheduled job with the ERD-rule. Thus, we have the following lemma.

Lemma 1. *For $P|r_j, reject|C_{\max} + \pi(R)$, there exists an optimal schedule such that the accepted jobs are processed in the ERD-rule on each machine.*

For convenience, for any $j \in \{1, 2, \dots, n\}$, write $u_j = r_j + p_j$ and let

$$U_j = \{J_i \in J | u_i > u_j\}$$

be the set of jobs with release time plus processing time larger than u_j . Correspondingly, let B_j be the set of jobs such that

$$B_j \supseteq U_j, \text{ and } \pi(B_j) \text{ is minimized.}$$

Then, the following lemma is obtained.

Lemma 2. For any $j \in \{1, 2, \dots, n\}$, B_j can be computed in polynomial time.

Proof. For each $j \in \{1, 2, \dots, n\}$, we can find the set U_j in polynomial time. Obviously, if $\pi(\cdot)$ is monotonically nondecreasing, then $B_j = U_j$ and the lemma holds.

Otherwise, we construct an auxiliary set function $w(\cdot)$ defined on all subsets of $J \setminus U_j$ as follows :

$$w(X) = \pi(X \cup U_j) - \pi(U_j), \quad X \subseteq J \setminus U_j. \quad (1)$$

By the submodularity of $\pi(\cdot)$, for any two subsets $X_1, X_2 \subseteq J \setminus U_j$, we have

$$\begin{aligned} & w(X_1) + w(X_2) \\ &= \pi(X_1 \cup U_j) - \pi(U_j) + \pi(X_2 \cup U_j) - \pi(U_j) \\ &\geq \pi((X_1 \cup U_j) \cup (X_2 \cup U_j)) + \pi((X_1 \cup U_j) \cap (X_2 \cup U_j)) - 2\pi(U_j) \\ &= \pi((X_1 \cup X_2) \cup U_j) - \pi(U_j) + \pi((X_1 \cap X_2) \cup U_j) - \pi(U_j) \\ &= w(X_1 \cup X_2) + w(X_1 \cap X_2). \end{aligned}$$

This implies that $w(\cdot)$ is a submodular function. Thus, $X' = \arg \min_{X: X \subseteq J \setminus U_j} \{w(X)\}$ can be computed within polynomial time using the method in [24]. Therefore, for any $X \in J \setminus U_j$, we have

$$\begin{aligned} w(X') &\leq w(X) \\ \pi(X' \cup U_j) - \pi(U_j) &\leq \pi(X \cup U_j) - \pi(U_j) \\ \pi(B_j) &\leq \pi(X \cup U_j), \end{aligned}$$

where $B_j = X' \cup U_j$. Thus, this lemma holds. \square

Let σ^* be the optimal schedule and let R^* be the rejected job set of σ^* . Write $A^* = J \setminus R^*$, $u_{j^*} = \max\{u_j | j \in A^*\}$ and $Z^* = C_{\max}(\sigma^*) + \pi(R^*)$, where $C_{\max}(\sigma^*)$ is the makespan of σ^* . Then, we have the following.

Lemma 3. There exists an optimal schedule σ^* that satisfies $B_{j^*} \subseteq R^*$.

Proof. Let $u_{j^*} = \max\{u_j | j \in J \setminus R^*\}$ and $U_{j^*} = \{J_i \in J | u_i > u_{j^*}\}$, then we have

$$U_{j^*} \subseteq R^*.$$

By Lemma 2, B_{j^*} can be found in polynomial time, where B_{j^*} is the set with minimum penalty satisfied $U_{j^*} \subseteq B_{j^*}$. This implies that

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

by $U_{j^*} \subseteq R^* \cap B_{j^*}$. Since $\pi(\cdot)$ is a submodular function, we have $\pi(R^*) + \pi(B_{j^*}) \geq \pi(R^* \cup B_{j^*}) + \pi(R^* \cap B_{j^*})$, i.e.,

$$\pi(R^*) \geq \pi(R^* \cup B_{j^*}).$$

Notably, assuming $B_{j^*} \setminus R^* \neq \emptyset$, we prove that there exists an optimal schedule σ' , in which all the jobs in $R^* \cup B_{j^*}$ are rejected.

Because the process time of any job is nonnegative, it follows that we can schedule all the jobs in $A^* \setminus B_{j^*}$ by schedule σ^* . This implies that the makespan $C_{\max}(\sigma')$ of the jobs in $A^* \setminus B_{j^*}$ is no more than $C_{\max}(\sigma^*)$. Thus, we have

$$Z^* = C_{\max}(\sigma^*) + \pi(R^*) \geq C_{\max}(\sigma') + \pi(R^* \cup B_{j^*}).$$

Therefore, σ' is an optimal schedule and this lemma holds. \square

3. Approximation Algorithm

In this section, we consider the problem $P|r_j, reject|C_{\max} + \pi(R)$ and propose a 2-approximation algorithm.

For each $j \in \{1, 2, \dots, n\}$, we introduce an auxiliary variable α_j , which is similar to the dual variable in the primal-dual method.

Lemma 4. Algorithm 1 can be implemented in polynomial time.

Algorithm 1 : \mathcal{A}

```

1 for each  $k \in \{1, 2, \dots, n\}$  do
2   using the method in [24] to compute the job set  $B_k$ .
3   Set the variable  $\alpha_j = 0$  for each  $J_j \in J$ ,  $F := B_k$  and  $R_k := B_k$ .
4   while  $J \setminus F \neq \emptyset$  do
5     Let

$$\Delta_1 = \min_{j: J_j \in J \setminus F} \left\{ \frac{p_j}{m} \right\};$$


$$\Delta_2 = \min_{S \subseteq J: S \setminus F \neq \emptyset} \left\{ \frac{\pi(S \cup B_k) - \pi(B_k) - \sum_{j: J_j \in S \cap F} \alpha_j}{\sum_{j: J_j \in S \setminus F} 1} \right\}.$$

6     if  $\Delta_1 \leq \Delta_2$  then
7       Freeze a job  $J_{j'}$  and set  $\alpha_{j'} := \Delta_1$ ,  $F := F \cup \{J_{j'}\}$ , where

$$j' = \arg \min_{j: J_j \in J \setminus F} \left\{ \frac{p_j}{m} \right\};$$

8     end
9     else
10      Freeze those unfrozen jobs in  $S'$  and set  $\alpha_j := \Delta_2$  for all jobs in  $S' \setminus F$ ,

$$R_k := R_k \cup S', F := F \cup S', \text{ where}$$


$$S' = \arg \min_{S: S \subseteq J, S \setminus F \neq \emptyset} \left\{ \frac{\pi(S \cup B_k) - \pi(B_k) - \sum_{j: J_j \in S \cap F} \alpha_j}{\sum_{j: J_j \in S \setminus F} 1} \right\};$$

11    end
12  Construct the schedule  $\sigma_k$  such that rejects the jobs in  $R_k$  and schedules all jobs
    in  $J \setminus R_k$  by the ERD-rule after time  $r_k$ . Let  $Z_k$  be the value of this schedule  $\sigma_k$ .
13 end
14 Among all the schedules obtained above, select one with the minimum  $Z_k$  value.
```

Proof. Clearly, B_k can be found in polynomial time by Lemma 2 for any $k \in \{1, 2, \dots, n\}$. Then, we consider the implementation of the while loops of Algorithm 1. Let $\alpha_j(t)$ and $F(t)$ be the value of dual variable of job J_j and the set of frozen jobs after the t -th execution of while loops of Algorithm 1, respectively. For convenience, we define $\alpha_j(0) = 0$, $J_j \in J$ and $F(0) = B_k$, $R_k(0) = B_k$. Note that the while loops of Algorithm 1 need to execute at most n times for any $k \in \{1, 2, \dots, n\}$.

For any t -th ($t \geq 1$) execution of the while loops of Algorithm 1, it is obvious that $\Delta_1(t) = \min_{j: J_j \in J \setminus F(t-1)} \left\{ \frac{p_j}{m} \right\}$ can be found in polynomial time for $t \in \{1, 2, \dots, n\}$.

Write

$$\begin{aligned} \Delta_2(t) &= \min_{S \subseteq J: S \setminus F(t-1) \neq \emptyset} \frac{\pi(S \cup B_k) - \pi(B_k) - \sum_{j: J_j \in S \cap F(t-1)} \alpha_j(t-1)}{\sum_{j: J_j \in S \setminus F(t-1)} 1} \\ &= \min_{S \subseteq J: S \setminus F(t-1) \neq \emptyset} \frac{w(S) + \alpha(S)}{k(S)}, \end{aligned}$$

where we define $w(S) = \pi(S \cup B_k) - \pi(B_k)$, $\alpha(S) = \sum_{j: J_j \in S \cap F(t-1)} (-\alpha_j(t-1))$ and $k(S) = \sum_{j: J_j \in S \setminus F(t-1)} 1$ for any subset $S \subseteq J$. Similar to the proof of Lemma 2, we have that $w(\cdot)$ is a submodular function. In particular, we can obtain that $w(\cdot) + \alpha(\cdot)$ is a submodular function because $\alpha(\cdot)$ and $k(\cdot)$ are linear functions. Then, using the combinatorial algorithm for the ratio of two submodular functions minimization problem considered in [25], the value of $\Delta_2(t)$ can be found in polynomial time. Thus, $S'(t) = \arg \min_{S: S \subseteq J, S \setminus F(t-1) \neq \emptyset} \left\{ \frac{\pi(S \cup B_k) - \pi(B_k) - \sum_{j: J_j \in S \cap F(t-1)} \alpha_j(t-1)}{\sum_{j: J_j \in S \setminus F(t-1)} 1} \right\}$ can be found in polynomial time.

Therefore, the lemma holds. \square

For any $k \in \{1, 2, \dots, n\}$, let σ_k be a feasible schedule in Algorithm 1 and let R_k be the rejected set of σ_k . In addition, let α_j be the value of dual variables when job J_j is frozen. Then, we have the following results because $\alpha_j = 0, \forall J_j \in B_k$ and the other α_j ($J_j \in J \setminus B_k$) can be obtained by the value $\min\{\Delta_1, \Delta_2\}$.

$$\begin{cases} \alpha_j = \frac{p_j}{m} \text{ for each job } J_j \in J \setminus R_k; \\ \alpha_j \leq \frac{p_j}{m} \text{ for each job } J_j \in J; \end{cases} \quad (2)$$

Additionally, we suppose that J_j is frozen at the \bar{t} -th execution of the while loops of Algorithm 1. Then, during the t -th implementation of the while loops of Algorithm 1, we have

$$\begin{cases} \alpha_j = 0, \text{ if } t < \bar{t} \\ \alpha_j(t) = \alpha_j, \text{ if } t \geq \bar{t}. \end{cases} \quad (3)$$

Moreover, we have the following results.

Lemma 5. For any $k \in \{1, 2, \dots, n\}$ and any $S \subseteq J$, we have

$$\sum_{j: J_j \in S} \alpha_j + \pi(B_k) \leq \pi(S \cup B_k). \quad (4)$$

Proof. For any $k \in \{1, 2, \dots, n\}$, we consider the t -th implementation of the while loops of Algorithm 1. We suppose that the number of the while loops of Algorithm 1 is T_k , i.e., $F(T_k) = J$.

For any $t \in \{1, 2, \dots, T_k\}$, assume that J_j is added to $F(t)$, we have

$$\alpha_j(t) = \min\{\Delta_1(t), \Delta_2(t)\} \leq \Delta_2(t), \quad (5)$$

and, for each subset $S \subseteq J$ with $S \setminus F(t-1) \neq \emptyset$, we have the following

$$\begin{aligned} \sum_{j: J_j \in S} \alpha_j(t) + \pi(B_k) &= \sum_{j: J_j \in S \setminus F(t-1)} \alpha_j(t) + \sum_{j: J_j \in S \cap F(t-1)} \alpha_j(t) + \pi(B_k) \\ &\leq \sum_{j: J_j \in S \setminus F(t-1)} \Delta_2(t) + \sum_{j: J_j \in S \cap F(t-1)} \alpha_j(t) + \pi(B_k) \\ &= \Delta_2(t) \sum_{j: J_j \in S \setminus F(t-1)} 1 + \sum_{j: J_j \in S \cap F(t-1)} \alpha_j(t) + \pi(B_k) \\ &\leq \frac{\pi(S \cup B_k) - \pi(B_k) - \sum_{j: J_j \in S \cap F(t-1)} \alpha_j(t)}{\sum_{j: J_j \in S \setminus F(t-1)} 1} \sum_{j: J_j \in S \setminus F(t-1)} 1 \\ &\quad + \sum_{j: J_j \in S \cap F(t-1)} \alpha_j(t) + \pi(B_k) \\ &= \pi(S \cup B_k), \end{aligned} \quad (6)$$

where the first inequality follows from relation (3) and inequality (5), i.e., $\alpha_j(t) = 0$ for $J_j \in S \setminus F(t)$ and $\alpha_j(t) = \Delta_2(t)$ for $J_j \in F(t) \setminus F(t-1)$, and the second inequality follows from the definition of $\Delta_2(t)$. We reach the conclusion of this lemma. \square

Lemma 6. For any $k \in \{1, 2, \dots, n\}$, the rejected job set R_k satisfies

$$\pi(R_k) = \sum_{j: J_j \in R_k} \alpha_j + \pi(B_k). \quad (7)$$

Proof. For any $k \in \{1, 2, \dots, n\}$, we consider the t -th implementation of the loops of Algorithm 1. Then, during the t' -th and t'' -th ($t' \leq t''$) implementation of the loops of Algorithm 1, we assume that $\Delta_2(t') < \Delta_1(t')$, $\Delta_2(t'') < \Delta_1(t'')$, let $S(t')$ and $S(t'')$ be the selected job sets in J at time t' and t'' , respectively. Thus, we have

$$\begin{aligned} \sum_{j: J_j \in S(t')} \alpha_j(t') + \pi(B_k) &= \sum_{j: J_j \in S(t') \setminus F(t')} \alpha_j(t') + \sum_{j: J_j \in S(t') \cap F(t')} \alpha_j(t') + \pi(B_k); \\ &= \sum_{j: J_j \in S(t') \setminus F(t')} \Delta_2(t') + \sum_{j: J_j \in S(t') \cap F(t')} \alpha_j(t') + \pi(B_k); \\ &= \frac{\pi(S(t') \cup B_k) - \pi(B_k) - \sum_{j: J_j \in S \cap F(t')} \alpha_j(t')}{\sum_{j: J_j \in S \setminus F(t')} 1} \sum_{j: J_j \in S(t') \setminus F(t')} 1 \\ &\quad + \sum_{j: J_j \in S(t') \cap F(t')} \alpha_j(t') + \pi(B_k); \\ &= \pi(S(t') \cup B_k), \end{aligned}$$

where B_k can be found by Algorithm 1, and then $\pi(B_k)$ is a constant. Similarly, we have

$$\sum_{j: J_j \in S(t'')} \alpha_j(t'') + \pi(B_k) = \pi(S(t'') \cup B_k).$$

For any job $J_j \in S(t')$, since $J_j \in S(t')$ is frozen at t' (or even earlier), we have $\alpha_j(t') = \alpha_j(t'')$ by relation (3), and

$$\begin{aligned} &\sum_{j: J_j \in S(t') \cup S(t'')} \alpha_j(t'') + \pi(B_k) + \sum_{j: J_j \in S(t') \cap S(t'')} \alpha_j(t'') + \pi(B_k) \\ &= \sum_{j: J_j \in S(t')} \alpha_j(t'') + \pi(B_k) + \sum_{j: J_j \in S(t'')} \alpha_j(t'') + \pi(B_k) \\ &= \pi(S(t') \cup B_k) + \pi(S(t'') \cup B_k) \\ &\geq \pi((S(t') \cup S(t'')) \cup B_k) + \pi((S(t') \cap S(t'')) \cup B_k) \\ &\geq \pi((S(t') \cup S(t'')) \cup B_k) + \sum_{j: J_j \in S(t') \cap S(t'')} \alpha_j(t'') + \pi(B_k), \end{aligned}$$

where the first inequality comes from the submodularity of $\pi(\cdot)$, and the second inequality follows by inequality (6).

This implies that

$$\sum_{j: J_j \in S(t') \cup S(t'')} \alpha_j(t'') + \pi(B_k) \geq \pi((S(t') \cup S(t'')) \cup B_k).$$

Inequality (6) and relation (3) indicate that

$$\sum_{j: J_j \in S(t') \cup S(t'')} \alpha_j + \pi(B_k) = \pi((S(t') \cup S(t'')) \cup B_k).$$

Therefore, we have $\pi(R_k) = \sum_{j:J_j \in R_k} \alpha_j + \pi(B_k)$ because R_k is equal to merging all subsets selected by this similar case, and then the lemma holds. \square

Let σ be the schedule obtained from Algorithm 1. Let Z and Z^* be the objective value of schedule σ and optimal schedule σ^* , respectively. Then, we have the following.

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

Proof. Let $C_{\max}(\sigma^*)$ be the makespan of σ^* and let R^* be the rejected job of σ^* ; then, the objective value of σ^* is

$$Z^* = C_{\max}(\sigma^*) + \pi(R^*).$$

Let $u_{j^*} = \max\{u_j | J_j \in J \setminus R^*\}$, by the pigeonhole principle, we have

$$C_{\max}(\sigma^*) \geq \sum_{J_j \in A^*} \frac{p_j}{m}, \quad (8)$$

where $A^* = J \setminus R^*$. By Lemma 3, without loss of generality, we assume that

$$B_{j^*} \subseteq R^*.$$

Consider $k = j^*$ during the implementation of Algorithm 1. We define σ_{j^*} and $C_{\max}(\sigma_{j^*})$ as the output schedule and the makespan of this schedule. Let R_{j^*} be the rejected job set of σ_{j^*} and let $\{\alpha_j\}_{J_j \in J}$ be the value of dual variables when all jobs in J are frozen. Then, we have

$$\begin{aligned} Z^* &= C_{\max}(\sigma^*) + \pi(R^*) \geq \sum_{J_j \in A^*} \frac{p_j}{m} + \pi(R^*) \\ &= \sum_{J_j \in A^*} \frac{p_j}{m} + \pi(R^* \cup B_{j^*}) \geq \sum_{J_j \in A^*} \frac{p_j}{m} + \sum_{j:J_j \in R^*} \alpha_j + \pi(B_{j^*}) \\ &= \sum_{J_j \in A^* \cap A_{j^*}} \frac{p_j}{m} + \sum_{J_j \in A^* \cap R_{j^*}} \frac{p_j}{m} + \sum_{j:J_j \in R^* \cap A_{j^*}} \alpha_j + \sum_{j:J_j \in R^* \cap R_{j^*}} \alpha_j + \pi(B_{j^*}) \\ &\geq \sum_{J_j \in A^* \cap A_{j^*}} \frac{p_j}{m} + \sum_{J_j \in A^* \cap R_{j^*}} \alpha_j + \sum_{j:J_j \in R^* \cap A_{j^*}} \frac{p_j}{m} + \sum_{j:J_j \in R^* \cap R_{j^*}} \alpha_j + \pi(B_{j^*}) \\ &= \sum_{j:J_j \in A_{j^*}} \frac{p_j}{m} + \pi(R_{j^*}), \end{aligned} \quad (9)$$

where $A_{j^*} = J \setminus R_{j^*}$, the first inequality follows from inequality (8), the second inequality follows from Lemma 5, the third inequality follows from inequality (2), and the last equality follows from Lemma 6.

Let $J_q \in A_{j^*}$ be the last completion job, i.e., the completion processed time of J_q is $C_{\max}(\sigma_{j^*})$. Furthermore, let s_q be the starting processing time of J_q in σ_{j^*} . Thus, we have $s_q \geq r_q$ and $C_{\max}(\sigma_{j^*}) = s_q + p_q$.

By the ERD-rule, if $s_q > r_q$, all machines are busy in the time interval $[r_q, s_q]$. Then, we have

$$m \cdot (s_q - r_q) \leq \sum_{j:J_j \in A_{j^*}} p_j \Rightarrow s_q - r_q \leq \sum_{j:J_j \in A_{j^*}} \frac{p_j}{m}.$$

Otherwise, for $s_q = r_q$, we have $s_q - r_q = 0 \leq \sum_{j:J_j \in A_{j^*}} \frac{p_j}{m}$ by $p_j \geq 0$ for each $J_j \in J$. Therefore, we have

$$s_q - r_q \leq \sum_{J_j \in A_{j^*}} \frac{p_j}{m}. \quad (10)$$

Furthermore, we have $r_q + p_q \leq u_{j^*} \leq C_{\max}(\sigma^*)$ because $J_q \in A_{j^*}$ and the definition of u_{j^*} . Then, summing up the inequalities (9) and (10), we can obtain the following

$$\begin{aligned} Z &\leq Z_{j^*} = C_{\max}(\sigma_{j^*}) + \pi(R_{j^*}) \\ &= s_q + p_q + \pi(R_{j^*}) \\ &= r_q + p_q + (s_q - r_q) + \pi(R_{j^*}) \\ &\leq C_{\max}(\sigma^*) + \sum_{J_j \in J_j \in A_{j^*}} \frac{p_j}{m} + \pi(R_{j^*}) \\ &\leq 2Z^*. \end{aligned}$$

To show that the bound is tight, we present an instance with four jobs and two parallel machines:

$$\begin{cases} J_1 = (r_1, p_1) = (0, 2); \\ J_2 = (r_2, p_2) = (2, 1); \\ J_3 = (r_3, p_3) = (2, 2); \\ J_4 = (r_4, p_4) = (4, 0). \end{cases}$$

and the submodular function is

$$\pi(S) = \begin{cases} 0, & |S| = \emptyset; \\ 6, & |S| = 1; \\ 9, & |S| = 2; \\ 11, & |S| = 3; \\ 12, & |S| = J = 4. \end{cases}$$

By Algorithm 1, when $k = 1$, the resulting schedule σ_1 is to reject all the jobs, and $Z_1 = \pi(J) = 12$; when $k = 2$ or $k = 3$, both the resulting schedule σ_2 and σ_3 are to reject J_4 and to process J_1 and J_2 on machine M_1 and process J_3 on machine M_2 , and $Z_1 = r_2 \text{ (or } 3) + C_{\max}(\sigma_2 \text{ (or } 3)) + \pi(\{J_4\}) = 2 + 4 + 6 = 12$; when $k = 4$, the resulting schedule σ_4 is to process J_1, J_2 and J_4 on machine M_1 and process J_3 on machine M_2 , and $Z_1 = r_4 + C_{\max}(\sigma_4) = 4 + 4 = 8$. The optimal schedule is to process J_1, J_2 and J_4 on machine M_1 and process J_3 on machine M_2 , and $Z^* = 4$. Thus, we have $Z + \min\{Z_1, Z_2, Z_3, Z_4\} = 8 = 2Z^*$.

Hence, we reach the conclusion of this theorem. \square

4. Conclusions

In this paper, we investigate parallel-machine scheduling with release times and submodular penalties ($P|r_j, reject|C_{\max} + \pi(R)$), which is a generalization of parallel-machine scheduling with release times and rejection penalties and single machine scheduling with release dates and submodular penalties. For $P|r_j, reject|C_{\max} + \pi(R)$, we propose a 2-approximation algorithm.

For parallel-machine scheduling with release times and rejection penalties, there exists a PTAS. For $P|r_j, reject|C_{\max} + \pi(R)$, there is a question of whether it is possible to design a PTAS or a further improved algorithm. Furthermore, establishing a better algorithm is an interesting direction for future work.

The vector scheduling problem [19,22,23] is a generalization of parallel machine scheduling, where each job J_j is associated with a d -dimensional vector. Thus, the vector parallel-machine scheduling with release times and rejection penalties, which can be viewed as one generalization of the $P|r_j, reject|C_{\max} + \pi(R)$, deserves to be explored. It is possible to design a 2-approximation algorithm, but it is a challenge.

In [26], Liu et al considered a k -prize-collecting cover problem, in which at least k points are covered. The k -prize-collecting scheduling problem with release times and rejection penalties, which can be viewed as another generalization of the $P|r_j, reject|C_{\max} + \pi(R)$, deserves to be explored.

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

Funding: The work is supported in part by the Project of Yunnan Provincial Department of Education Science Research Fund (NO. 2019Y0022).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Graham, R.L. Bounds on multiprocessing timing anomalies. *SIAM J. Appl. Math.* **1969**, *17*, 416–429. [\[CrossRef\]](#)
2. Davis, E.; Jaffe, J.M. Algorithms for scheduling tasks on unrelated processors. *J. ACM* **1981**, *28*, 721–736. [\[CrossRef\]](#)
3. Hochbaum, D.S.; Shmoys, D.B. Using dual approximation algorithms for scheduling problems theoretical and practical results. *J. ACM* **1987**, *34*, 144–162. [\[CrossRef\]](#)
4. Hochbaum, D.S.; Shmoys, D.B. Approximation algorithms for scheduling unrelated parallel machines. *Math. Program.* **1990**, *46*, 259–271.
5. Bartal, Y.; Leonardi, S.; Marchetti-Spaccamela, A.; Sgall, J.; Stougie, L. Multiprocessor scheduling with rejection. *SIAM J. Discret. Math.* **2000**, *13*, 64–78. [\[CrossRef\]](#)
6. Ou, J.; Zhong, X.; Wang, G. An improved heuristic for parallel machine scheduling with rejection. *Eur. J. Oper. Res.* **2015**, *241*, 653–661. [\[CrossRef\]](#)
7. Zhang, L.; Lu, L.; Yuan, J. Single machine scheduling with release dates and rejection. *Eur. J. Oper. Res.* **2009**, *198*, 975–978. [\[CrossRef\]](#)
8. Zhang, L.; Lu, L. Parallel-machine scheduling with release dates and rejection. *4OR-A Q. J. Oper. Res.* **2016**, *14*, 387–406. [\[CrossRef\]](#)
9. 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. [\[CrossRef\]](#)
10. Zhong, X.; Pan, Z.; Jiang, D. Parallel-machine scheduling with release dates and rejection. *J. Comb. Optim.* **2016**, *33*, 934–944. [\[CrossRef\]](#)
11. Fujishige, S. *Submodular Functions and Optimization*, 2nd ed.; Elsevier: Amsterdam, The Netherlands, 2005.
12. Edmonds, J. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Optimization*; Jünger, M., Reinelt, G., Rinaldi, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2003.
13. Du, D.; Lu, R.; Xu, D. A Primal-dual approximation algorithm for the facility location problem with submodular penalties. *Algorithmica* **2012**, *63*, 191–200. [\[CrossRef\]](#)
14. Liu, X.; Li, W. Approximation algorithms for the submodular load balancing with submodular penalties. *Optim. Lett.* **2021**, *15*, 2165–2180. [\[CrossRef\]](#)
15. 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. [\[CrossRef\]](#)
16. Liu, X.; Li, W. Approximation algorithm for the single machine scheduling problem with release dates and submodular rejection penalty. *Mathematics* **2020**, *8*, 133. [\[CrossRef\]](#)
17. Liu, X.; Li, W. Combinatorial approximation algorithms for the submodular multicut problem in trees with submodular penalties. *J. Comb. Optim.* **2020**. [\[CrossRef\]](#)
18. Liu, X.; Xing, P.; Li, W. Approximation algorithms for the submodular load balancing with submodular penalties. *Mathematics* **2020**, *8*, 1785. [\[CrossRef\]](#)
19. Liu, X.; Li, W.; Zhu, Y. Single machine vector scheduling with general penalties. *Mathematics* **2021**, *9*, 1965. [\[CrossRef\]](#)
20. Guan, L.; Li, W.; Xiao, M. Online algorithms for the mixed ring loading problem with two nodes. *Optim. Lett.* **2021**, *15*, 1229–1239. [\[CrossRef\]](#)
21. Li, W.; Li, J.; Zhang, X.; Chen, Z. Penalty cost constrained identical parallel machine scheduling problem. *Theor. Comput. Sci.* **2015**, *607*, 181–192. [\[CrossRef\]](#)
22. Li, W.; Cui, Q. Vector scheduling with rejection on a single machine. *4OR-A Q. J. Oper. Res.* **2018**, *16*, 95–104. [\[CrossRef\]](#)

23. Dai, B.; Li, W. Vector scheduling with rejection on two machines. *Int. J. Comput. Math.* **2020**, *97*, 2507–2515. [[CrossRef](#)]
24. Iwata, S.; Fleischer, L.; Fujishige, S. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *J. ACM* **2001**, *48*, 761–777. [[CrossRef](#)]
25. 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](#)]
26. Liu, X.; Li, W.; Xie, R. A primal-dual approximation algorithm for the k -prize-collecting minimum power cover problem. *Optim. Lett.* **2021**. [[CrossRef](#)]