Approximation Algorithms for the Submodular Load Balancing with Submodular Penalties

Xiaofei Liu 1,*, Peiyin Xing 1 and Weidong Li 2,*

1 School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China; pyxing@pku.edu.cn
2 School of Mathematics and Statistics, Yunnan University, Kunming 650504, China
* Correspondence: xiaofei@pku.edu.cn (X.L.); weidong@ynu.edu.cn (W.L.)

Received: 24 August 2020; Accepted: 12 October 2020; Published: 15 October 2020

Abstract: In this paper, we study the submodular load balancing problem with submodular penalties. The objective of this problem is to balance the load among sets, while some elements can be rejected by paying some penalties. Officially, given an element set \( V \), we want to find a subset \( R \) of rejected elements, and assign other elements to one of \( m \) sets \( A_1, A_2, \ldots, A_m \). The objective is to minimize the sum of the maximum load among \( A_1, A_2, \ldots, A_m \) and the rejection penalty of \( R \), where the load and rejection penalty are determined by different submodular functions. We study the submodular load balancing problem with submodular penalties under two settings: heterogenous setting (load functions are not identical) and homogenous setting (load functions are identical). Moreover, we design a Lovász rounding algorithm achieving a worst-case guarantee of \( m + 1 \) under the heterogenous setting and a \( \min\{m, \left\lceil \frac{n}{m} \right\rceil + 1 \} = O(\sqrt{n}) \)-approximation combinatorial algorithm under the homogenous setting.

Keywords: load balancing; submodular function; submodular penalties; approximation algorithm

1. Introduction

The load balancing problem, also called the minimum makespan scheduling problem, is one of the most classical and active topics in the field of combinational optimization and approximation algorithms that have been widely studied for more than five decades. The load balancing problem is to find an optimal \( m \)-partition assigning \( n \) elements to \( m \) sets so as to minimize the maximum load among all sets, where \( p_{ij} \) is the load for element \( j \) to be assigned to the set \( i \). If \( p_{ij} = p_{i'j} \) for any \( 1 \leq i < i' \leq m \), we call it load balancing under the homogeneous setting. Otherwise, we call it load balancing under the heterogeneous setting.

Under the homogeneous setting, Graham [1] proved that load balancing is strong \( NP \)-hard, and achieved 2-approximation using the classical list scheduling (LS, for short) algorithm, which was improved by himself with a 4/3-approximation algorithm [2]. When \( m \) is a given constant, Jansen and Porkolab [3] presented a fully polynomial-time approximation scheme (FPTAS, for short) running in time \( O(n(m/\varepsilon)^{O(m)}) \), which can be improved to \( O(n) + (1/\varepsilon)^{O(m)} \) when \( 1/\varepsilon > m \) [4]. When \( m \) is an arbitrary input, Hochbaum and Shmoys [5] designed a polynomial-time approximation scheme (PTAS, for short), which can give an \( (1 + \varepsilon) \)-approximation for any \( \varepsilon > 0 \). PTAS can be further improved to an efficient polynomial-time approximation scheme (EPTAS, for short). Alon et al. [6], Jansen et al. [7,8] designed EPTAS for this problem and [8] provided the best result among them.
Under the heterogeneous setting, when \( m \) is a given constant, Horowitz and Sahni [9] designed an FPTAS running in time \( O(n m (n m / \epsilon)^{m-1}) \) requiring space \( O((n m / \epsilon)^{m-1}) \). The required space was reduced by Lenstra et al. [10], which is bounded by \( m \) and \( \log(1/\epsilon) \). This paper also proved that when \( m \) is an arbitrary input, this problem cannot be approximated within \( 3/2 - \epsilon \) for any \( \epsilon > 0 \) and provided a 2-approximation algorithm using the LP relaxation algorithm. Recently, Kones and Levin [11] devised a unified approximation framework for designing EPTAS for load balancing on parallel machines. Besides, Bhaskara et al. [12] and Chen et al. [13] studied a special case when the rank of the matrix formed by the load for any element is small.

All elements have to be assigned in load balancing. However, assign all elements would be too costly or even impossible in the real world due to resource limitations or certain policies. With this motivation, Bartal et al. [14] proposed the multiprocessor scheduling with a rejection, where an element is either rejected, in which case a rejection penalty has to be paid, or accepted and assigned to the set. The objective changes slightly to minimize the maximum load among all sets plus the sum of the penalties of the rejected elements. For multiprocessor scheduling with a rejection under the homogeneous setting, Bartal et al. [14] proposed a 2-approximation algorithm, which is improved by Ou et al. [15] with a \( 3/2 + \epsilon \)-approximation algorithm. More related results can be found in [16–18].

In recent years, submodular load balancing, which is a generalization of load balancing, is investigated. Submodular functions have the property of decreasing marginal returns which is verified in mathematical models in various fields such as operations research, economics, engineering, computer science, management science, etc. They are extensively utilized to solve practical problems in image segmentation, parallel computing, viral marketing and data summarization [19–22]. Since the submodular property covers a much wider range of functions compared with modular property, problems with submodular settings are often harder to cope with. Even in the homogeneous setting of load balancing problem, Svitkina and Fleischer [23] showed that submodular load balancing is information-theoretically hard to approximate within \( o(\sqrt{n} / \log n) \). Using a similar method, Wei et al. [22] showed that submodular load balancing is information-theoretically hard to approximate within \( m \) if \( m \ll n \). Svitkina and Fleischer [23] provided a balanced partitioning algorithm yielding a factor of \( \min\{m, n/m\} \) and a sampling-based algorithm achieving \( O(\sqrt{n} / \log n) \) under the homogeneous setting, respectively. Using ellipsoid approximation techniques, Goemans et al. [24] presented an \( O(\sqrt{n} \log n) \)-approximation algorithm for submodular load balancing under the homogeneous setting. Using Lovász rounding algorithm, Wei et al. [22] presented an \( m \)-approximation algorithm for submodular load balancing under the homogeneous setting. Meanwhile, combinatorial optimization problems with submodular penalties are receiving increasing attentions [25–29].

Most recently, Zhang et al. [30] proposed a 3-approximation algorithm for precedence-constrained scheduling with submodular rejection on parallel machines. Liu and Li [31] proposed a 2-approximation algorithm for single machine scheduling with releasing time and submodular rejection.

In this paper, we consider the submodular load balancing with submodular penalties, which is a combination of multiprocessor scheduling with rejection and submodular load balancing, where the load and rejection penalty are all determined by submodular functions. Compared with the submodular load balancing, our problem is closer to the real situation because some elements can be rejected. Besides, in the real world, there is no linear relationship between the number of rejected elements and the rejection penalties which can be considered as the loss of the manufacturer’s prestige, but as the number of rejected elements increases, it will have fewer and fewer rejection penalties. Therefore, the submodular load balancing with submodular penalties is closer to reality than all problems introduced above.

We have dealt with this problem with two settings (heterogenous and homogenous) in this paper. As shown in Table 1, under the heterogeneous setting, we design a non-combinational Lovász rounding algorithm achieving a worst-case guarantee of \( m + 1 \). Under the homogeneous setting, we design an \( O(\sqrt{n}) \)-approximation combinational algorithm by combining two approximation algorithms,
one is an \( m \)-approximation combinational algorithm based on the greedy technique and another is \( \lceil \frac{n}{m} \rceil + 1 \)-approximation combinational algorithm based on the average rule.

### Table 1. The results in this paper.

| Case                                      | Setting               | Our Result              |
|-------------------------------------------|-----------------------|-------------------------|
| The submodular load balancing with        | Heterogeneous         | \( m + 1 \) (non-combinational) |
| submodular penalties                     | Homogenous            | \( \min \{m, \lceil \frac{n}{m} \rceil + 1 \} = O(\sqrt{n}) \) (combinational) |

The rest of this paper is structured as follows. In Section 2, we provide basic definitions and a formal problem statement. In Section 3, we focus on the heterogeneous setting and propose a Lovász rounding algorithm. In Section 4, we further consider the homogenous setting and provide a combinatorial \( m \)-approximation algorithm. In Section 5, we give a brief conclusion.

### 2. Preliminaries

Let \( V = \{v_1, v_2, \ldots, v_n\} \) be a given ground set of \( n \) elements, and \( f(\cdot) : 2^V \to \mathbb{R}_{\geq 0} \) be a real-valued function defined on all the subsets of \( V \). The following definitions are used frequently in this paper.

- \( f(\cdot) \) is called submodular if \( f(S) + f(T) \geq f(S \cup T) + f(S \cap T), \forall S, T \subseteq V \).
- \( f(\cdot) \) is called nondecreasing if \( f(S) \leq f(T) \) for any \( S \subseteq T \subseteq V \).
- \( f(\cdot) \) is called normalized if \( f(\emptyset) = 0 \).
- \( f(\cdot) \) is called polymatroid if \( f(\cdot) \) is normalized, nondecreasing and submodular.

The submodular load balancing (SLB, for short) with submodular penalties can be described as follows. Given a ground set \( V = \{v_1, v_2, \ldots, v_n\}, m \) \( (\leq n) \) polymatroid load functions \( f_1(\cdot), \ldots, f_m(\cdot) : 2^V \to \mathbb{R}_{\geq 0} \), and a polymatroid rejection function \( \pi(\cdot) : 2^V \to \mathbb{R}_{\geq 0} \). The SLB with submodular penalties is to choose a partition \( (A_1, \ldots, A_m, R) \) of the ground set \( V \), where \( A = \bigcup_{i=1}^{m} A_i \) is the set of accepted elements, \( R \) is the set of rejected elements, and

\[
 A_{i_1} \cap A_{i_2} = \emptyset, \quad \forall 1 \leq i_1 < i_2 \leq n, \quad \text{and} \quad A \cup R = V.
\]

The objective value of partition \( (A_1, \ldots, A_m, R) \) is \( \max_i f_i(A_i) + \pi(R) \), where \( f_i(A_i) \) is the load of the \( i \)th partition. The objective is to find a partition \( (A_1, \ldots, A_m, R) \) with the minimum objective value.

For each subset \( S \subseteq V \) and \( i \in \{1, 2, \ldots, m\} \), we introduce a binary variable \( x_{i,S} \), where

\[
 x_{i,S} = \begin{cases} 
 1, & \text{if } A_i = S; \\
 0, & \text{otherwise}, 
\end{cases}
\]

For each subset \( S \subseteq V \), we introduce a binary variable \( z_S \), where

\[
 z_S = \begin{cases} 
 1, & \text{if } S = R; \\
 0, & \text{otherwise}. 
\end{cases}
\]

Correspondingly, the load of \( A_i \) is defined as

\[
 f_i(A_i) = \sum_{S \subseteq V} f_i(S)x_{i,S}.
\]
Thus, the integer linear program (ILP) for the SLB with submodular penalties can be formulated as:

\[
\min \left( \max_i \sum_{S \subseteq V} f_i(S)x_{i,S} + \sum_{S \subseteq V} \pi(S)z_S \right)
\]

s.t.
\[
\sum_{i=1}^{m} \sum_{S \subseteq V \text{ and } v_i \in S} x_{i,S} + \sum_{S \subseteq V \text{ and } v_i \in S} z_S \geq 1, \forall j = 1, 2, \ldots, n, \\
x_{i,S}, z_S \in \{0,1\}, \forall i = 1, 2, \ldots, m, \forall S \subseteq V.
\]

where \(f_1(\cdot), \ldots, f_m(\cdot)\) and \(\pi(\cdot)\) are polymatroid functions as mentioned above.

Clearly, if \(\pi(S) = +\infty\) for any \(S \neq \emptyset\), ILP(1) is exactly the SLB problem considered in [22]. If \(\pi(S) = \sum_{v_j \in S} \pi(\{v_j\})\) and \(f_i(S) = \sum_{v_j \in S} f(\{v_j\})\) for any \(S \neq \emptyset\) and \(i = 1, 2, \ldots, m\), ILP(1) is exactly multiprocessor scheduling with rejection considered in [14]. Therefore, the SLB with submodular penalties generalizes both the problems in [14,22]. We further categorize the SLB with submodular penalties depending on whether the \(f_1(\cdot), \ldots, f_m(\cdot)\) are identical to each other (homogeneous), i.e., \(f_i(S) = f(S)\) for any \(i, S\), or not (heterogeneous).

As mention above, SLB with submodular penalties generalizes the SLB problem [22,23]. Thus,

**Lemma 1.** [23] The submodular load balancing with submodular penalties is hard to approximate to a factor of \(o(\sqrt{n/\log n})\) even for the homogeneous setting.

If \(m \ll n\), we have

**Corollary 1.** [22] For any \(\varepsilon > 0\), the submodular load balancing with submodular penalties cannot be approximated to a factor of \((1 - \varepsilon)m\) for any \(m = o(\sqrt{n/\log n})\) with polynomial number of queries even for the homogeneous setting.

In the remainder of this paper, we propose a \((m + 1)\)-approximation algorithm using the Lovász extension for the problem under the heterogeneous setting. Then, we design a combination \(O(\sqrt{n})\)-approximation algorithm for the problem under the homogeneous setting.

### 3. Heterogeneous Setting

In this section, we first introduce the definition of Lovász extension to show that the optimal relaxation solution of (1) can be solved in polynomial time. Then, round the optimal relaxation solution to a feasible solution of the SLB with submodular penalties under the heterogeneous setting.

Let \(V = \{v_1, v_2, \ldots, v_n\}\). For each vector \(x \in \mathbb{R}_{\geq 0}^n\), it is represented as

\[x = (x(1), x(2), \ldots, x(n)) \in \mathbb{R}_{\geq 0}^n.\]

For each subset \(S \subseteq V\), let \(I_S = (I_S(1), I_S(2), \ldots, I_S(n)) \in \mathbb{R}_{\geq 0}^n\) be the characteristic vector of set \(S\), where

\[I_S(j) = \begin{cases} 1, & \text{if } v_j \in S, \\ 0, & \text{otherwise.} \end{cases}\]

**Definition 1.** (Lovász extension [32]) The Lovász extension of a normalized submodular function \(f(\cdot)\) is a function \(\hat{f}(\cdot) : \mathbb{R}_{\geq 0}^n \rightarrow \mathbb{R}_{\geq 0}\) satisfying:
1. For each subset $S \subseteq V$, $\hat{f}(I_S) = f(S)$.
2. For any vector $x = (x(1), x(2), \ldots, x(n)) \in \mathbb{R}_{\geq 0}^n$
\[
\hat{f}(x) = \max \sum_{j=1}^{n} \alpha_j \cdot x(j)
\]
s.t. \[
\sum_{v_j \in S} \alpha_j \leq f(S), \ \forall S \subset V,
\]
\[
\alpha_j \geq 0,
\]

The following theorem reveals the relationship between set function and convexity.

**Theorem 1.** [32] A set function $f(\cdot)$ is submodular if and only if its Lovász extension $\hat{f}(\cdot)$ is a convex function.

Without loss of generality, we assume that $x(1) \geq x(2) \geq \cdots \geq x(n)$, and let $S_j = \{v_1, v_2, \ldots, v_j\}$ for any $j = 1, 2, \ldots, n$ and $S_0 = \emptyset$. From Lemma 3.1 in [33], we can construct an optimal solution $\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_n^*)$ of (2), where
\[
\alpha_j^* = f(S_j) - f(S_{j-1}).
\]

Consider a feasible partition $(A_1, \ldots, A_m, R)$. For each element $v_j \in V$, we introduce a binary variable $z(j)$, where $z(j) = 1$ if and only if $v_j$ is rejected. For every element $v_j$ and every $A_i$, we introduce a binary variable $x_i(j)$ where $x_i(j) = 1$ if and only if $v_j \in A_i$. For each $A_i$, by the definition of Lovász extension, we have
\[
\hat{f}_i(I_{A_i}) = f_i(A_i) = \sum_{S: S \subseteq V} f_i(S) x_{i,S},
\]
where $x_{i,S} = 1$ if and only if $S = A_i$. Similarly, for the rejected set $R$, we have
\[
\hat{\pi}(I_R) = \pi(R) = \sum_{S: S \subseteq V} \pi(S) z_S,
\]
where $I_R$ is the characteristic vector of $R$ and $z_S = 1$ if and only if $S = R$. Therefore, ILP(1) is equivalent to the following convex programming:
\[
\min \left( \max_i \hat{f}_i(x_i) + \hat{\pi}(z) \right)
\]
s.t. \[
\sum_{i=1}^{m} x_i(j) + z(j) \geq 1, \ \forall j = 1, 2, \ldots, n,
\]
$x_i(j), z(j) \in \{0, 1\}$, $\forall i = 1, 2, \ldots, m$
and $\forall j = 1, 2, \ldots, n,$
The relaxed version is

\[
\min \left( \max_i \hat{f}_i(x_i) + \hat{\pi}(z) \right)
\]

s.t. \[
\sum_{i=1}^{m} x_i(j) + z(j) \geq 1, \ \forall j = 1, 2, \ldots, n,
\]

\( \sum_{i=1}^{m} x_i(j), z(j) \in [0, 1], \ \forall i = 1, 2, \ldots, m \)

and \( \forall j = 1, 2, \ldots, n. \) (3)

Let \((x_1^*, x_2^*, \ldots, x_m^*, z^*)\) be the optimal solution of the convex relaxation (3), which can be solved in polynomial time by the ellipsoid method. We design a \((m + 1)\)-approximation algorithm by the rounding technique, which is described as follows.

**Theorem 2.** Algorithm 1 achieves a worst-case guarantee of \(m + 1\).

---

**Algorithm 1:** Lovász rounding algorithm.

1. Initially, set \(R = \emptyset\) and \(A_i = \emptyset\) for any \(i = 1, 2, \ldots, m\), where \(R\) denotes the rejected set and \(A_1, \ldots, A_m\) denote a partition of \(V \setminus R\), respectively.
2. Find an optimal solution \((x_1^*, x_2^*, \ldots, x_m^*, z^*)\) of the convex relaxation (3).
3. for \(j = 1, 2, \ldots, n\) do
   4. if \(z^*(j) \geq \frac{1}{m+1}\) then
      5. Add \(v_j\) to rejected set \(R\).
   6. else
      7. Set \(\hat{i} \in \arg \max_i x_i^*(j)\), and add \(v_j\) to \(A_{\hat{i}}\).
8. Output \((A_1, \ldots, A_m, R)\).

**Proof.** For the vector \(z^*\), without loss of generality, we assume

\[
z^*(1) \geq z^*(2) \geq \cdots \geq z^*(n).
\]

For each element \(v_j \in V\), if \(z^*(j) \geq \frac{1}{m+1}\), \(v_j\) is rejected and \(v_j \in R\), according to Algorithm 1. It implies that \(I_R(j) = 1\), and

\[
z^*(j) \geq \frac{1}{m+1} = \frac{1}{m+1} I_R(j).
\]

If \(z^*(j) < \frac{1}{m+1}\), we have \(I_R(j) = 0\). Thus, for any case, we have

\[
z^*(j) \geq \frac{1}{m+1} I_R(j), \ \forall j = 1, 2, \ldots, n.
\] (4)
By the definition of Lovász extension, we have

\[ \hat{\pi}(z^*) = \sum_{j=1}^{n-1} (z^*(j) - z^*(j+1)) \pi(S_j) + z^*(n) \pi(S_n) \]

\[ = z^*(1) \pi(S_1) + \sum_{j=2}^{n} z^*(j) (\pi(S_j) - \pi(S_{j-1})) \]

\[ \geq \frac{1}{m+1} I_R(1) \pi(S_1) \]

\[ + \sum_{j=2}^{n} \frac{1}{m+1} I_R(j) (\pi(S_j) - \pi(S_{j-1})) \]

\[ = \hat{\pi}(\frac{1}{m+1} I_R) \]

where \( S_j = \{v_1, v_2, \ldots, v_j\} \) for any \( j = 1, 2, \ldots, n \), the inequality follows from inequality (4) and the fact that \( \pi(\cdot) \) is the nondecreasing, and the last equality follows from the definition of \( \hat{\pi} \). Therefore, we have

\[ \pi(R) = \hat{\pi}(I_R) = (m+1) \hat{\pi}(\frac{1}{m+1} I_R) \]

\[ \leq (m+1) \hat{\pi}(z^*). \quad (5) \]

For any \( v_j \in V \backslash R \), we have \( z^*(j) < \frac{1}{m+1} \), by the choice of Algorithm 1. Therefore,

\[ \max_i x_i^*(j) \geq \frac{1}{m} \sum_{i=1}^{m} x_i^*(j) \geq \frac{1}{m} \sum_i (1 - z^*(j)) > \frac{1}{m+1}, \]

where the second inequality follows from that \( x_i^* = (x_i^*(1), \ldots, x_i^*(n)) \) is the optimal solution of the convex relaxation (3). Similarly, for every \( i = 1, 2, \ldots, m \), we have

\[ f_i(A_i) = \hat{f}_i(I_{A_i}) = (m+1) \hat{f}_i(\frac{1}{m+1} I_{A_i}) \]

\[ \leq (m+1) \hat{f}_i(x_i^*). \quad (6) \]

Therefore, we have

\[ \max_i f_i(A_i) + \pi(R) \]

\[ \leq (m+1) \max_i f_i(x_i^*) + (m+1) \hat{\pi}(z^*) \]

\[ = (m+1) \cdot (\max_i f_i(x_i^*) + \hat{\pi}(z^*)) \]

\[ \leq (m+1) \cdot \text{OPT}, \]

where the first inequality follows from inequalities (5) and (6), and \( \text{OPT} \) is the optimal value of the ILP (1).

4. Homogeneous Setting

In this section, we present two approximation algorithms to solve the SLB with submodular penalties under the homogeneous setting. We begin by presenting an \( m \)-approximation combinational algorithm based on the greedy technique. Then, we give a \( \lceil \frac{m}{m+1} \rceil + 1 \)-approximation combinational algorithm based on
the average partition of the elements into \( A_1, \ldots, A_m, R \). Combining these two approximation algorithms, we can design a \( O(\sqrt{n}) \)-approximation combinational.

For the SLB with submodular penalties under the homogeneous setting, we have that \( f_1(\cdot), \ldots, f_m(\cdot) \) are identical. For convenience, we use polymatriod load function \( f(\cdot) \) to represent each load function \( f_1(\cdot), \ldots, f_m(\cdot) \), i.e., for any \( i = 1, \ldots, m \),

\[
f(S) = f_i(S), \forall S \subseteq V.
\]

For this problem, we propose a combinatorial \( m \)-approximation algorithm using an auxiliary set function \( w(\cdot) \) with the ground set \( V \), where

\[
w(S) = f(V \setminus S) + \pi(S), \forall S \subseteq V. \tag{7}
\]

We propose the detailed \( m \)-approximation combinational algorithm as follows.

**Theorem 3.** Algorithm 2 achieves a worst-case guarantee of \( m \) in polynomial time.

**Algorithm 2:** Algorithm based on the greedy technique.

1. Initially, set \( R = A_i = \emptyset \) for any \( i = 1, 2, \ldots, m \).
2. Construct set function \( w(\cdot) \) as defined in (7).
3. Set \( R \in \arg\min_{S \subseteq V} w(S) \).
4. for \( v_j \in V \setminus R \) do
5. Set \( i \in \arg\min_i f(A_i) \), and add \( v_j \) to \( A_i \).
6. Output \((A_1, \ldots, A_m, R)\).

**Proof.** Since \( f(\cdot) \) and \( \pi(\cdot) \) are polymatriod, for any set \( S, T \subseteq V \), we have

\[
\begin{align*}
w(S) + w(T) &= f(V \setminus S) + \pi(S) + f(V \setminus T) + \pi(T) \\
&\geq f((V \setminus S) \cup (V \setminus T)) + f((V \setminus S) \cap (V \setminus T)) \\
&\quad + \pi(S \cup R) + \pi(S \cap R) \\
&= f(V \setminus (S \cap T)) + \pi(S \cap R) \\
&\quad + f(V \setminus (S \cup T)) + \pi(S \cup R) \\
&= w(S \cap T) + w(S \cup T),
\end{align*}
\]

where the inequality follows from the submodularity of \( f(\cdot) \) and \( \pi(\cdot) \). It implies that \( w(\cdot) \) is submodular. By modifying the method in [34] slightly, \( \min_{S \subseteq V} w(S) \) and corresponding solution \( R \) can be computed within polynomial time. It is easy to verify other processes can be implemented in polynomial time.

Let \((A_1^*, \ldots, A_m^*, R^*)\) be the optimal partition for the SLB with submodular penalties under the homogeneous setting. Following from the definition of \( R \), we have

\[
\begin{align*}
f(V \setminus R) + \pi(R) &= w(R) \leq w(R^*) \\
&= f(V \setminus R^*) + \pi(R^*). \tag{8}
\end{align*}
\]
Therefore, the objective value of \((A_1, A_2, \ldots, A_m, R)\) produced by Algorithm 2 is
\[
\max_i f(A_i) + \pi(R) \leq f(V \setminus R) + \pi(R)
\]
\[
\leq f(V \setminus R^*) + \pi(R^*)
\]
\[
\leq \sum_i f(A_i^\ast) + \pi(R^*)
\]
\[
\leq m(\max_i f(A_i^\ast) + \pi(R^*))
\]
\[
= m\text{OPT},
\]
where the first inequality follows since \(f(\cdot)\) is nondecreasing, the second inequality follows from the inequality (8), and the third inequality follows from the submodularity of \(f(\cdot)\).

Then, we present an \((\lceil \frac{n}{m} \rceil + 1)\)-approximation algorithm for the SLB with submodular penalties under the homogeneous setting. Consider the following decision version of the problem: given an instance \(I\) and target load \(l\) and target penalty \(\tau\), does there exist a partition with its load at most \(l\) and its rejected penalty at most \(\tau\). On input \((I, l, \tau)\), the algorithm decision procedure would output ‘Fail’ or ‘\((A_1, \ldots, A_m, R)\)’, depending on, whether there was such a partition. The details of the proposed algorithm are shown as follows.

Let \((A_1^\ast, \ldots, A_m^\ast, R^\ast)\) be the optimal partition for the SLB with submodular penalties under the homogeneous setting and
\[
a^\ast = \max_i f(A_i^\ast) \text{ and } r^\ast = \pi(R^\ast)
\]

**Theorem 4.** For \(a = a^\ast\) and \(r = r^\ast\), Algorithm 3 achieves a feasible partition with objective value at most \((\lceil \frac{n}{m} \rceil + 1)(a^\ast + r^\ast)\).

**Algorithm 3:** Algorithm based on the average rule.

1. Initially, set \(R = A_i = \emptyset\) for any \(i = 1, 2, \ldots, m\).
2. for \(v_j \in V\) do
3. if \(f(\{v_j\}) > a\) and \(\pi(\{v_j\}) > r\) then
4. Fail.
5. if \(f(\{v_j\}) > a\) then
6. Add \(v_j\) to \(R\).
7. if \(\pi(\{v_j\}) > r\) then
8. Set \(i \in \arg\min_i |A_i|\), and add \(v_j\) to \(A_i\).
9. for \(v_j \in V \setminus (\bigcup_i (A_i) \cup R)\) do
10. Add \(v_j\) to \(S\), where \(S\) is a set among \(A_1, \ldots, A_m, R\) with the minimum number of elements.
11. Output \((A_1, \ldots, A_m, R)\).

**Proof.** Let \((A_1, \ldots, A_m, R)\) be the output partition by Algorithm 3 and
\[
\begin{align*}
A_L &= \{v_j \in A | \pi(\{v_j\}) > r^\ast\}; \\
R_L &= \{v_j \in R | f(\{v_j\}) > a^\ast\}.
\end{align*}
\]

For each element \(v_j \in R_L\), we have \(v_j \in R^\ast\), otherwise \(\max_i f(A_i^\ast) > a^\ast\). It implies that \(R_L \subseteq R^\ast\) and
\[
\pi(R_L) \leq \pi(R^\ast) = r^\ast
\]
that \( \pi(\cdot) \) is nondecreasing.

For each \( v_j \in A_L \), \( v_j \) is the partition to a set among \( A_1, \ldots, A_m \) with the minimum number of elements. Moreover, for each \( v_j \in V \setminus (A_L \cup R_L) \) is the partition to a set among \( A_1, \ldots, A_m, R \) with the minimum number of elements. Thus, we claim

\[
\max_i |A_i| \leq \left\lceil \frac{|A_L|}{m} \right\rceil + \left\lceil \frac{|V \setminus (A_L \cup R_L)|}{m+1} \right\rceil \\
\leq \left\lceil \frac{|A_L|}{m} \right\rceil + \left\lceil \frac{|V \setminus (A_L \cup R_L)|}{m+1} \right\rceil + 1 \\
\leq \left\lceil \frac{n}{m} \right\rceil + 1
\]

and

\[
|R| \leq |R_L| + \left\lceil \frac{|V \setminus (A_L \cup R_L)|}{m+1} \right\rceil \\
\leq |R_L| + \left\lceil \frac{n}{m+1} \right\rceil \\
\leq |R_L| + \left\lceil \frac{n}{m} \right\rceil.
\]

Therefore, the objective value of \((A_1, A_2, \ldots, A_m, R)\) produced by Algorithm 3 is

\[
\max_i f(A_i) + \pi(R) \\
\leq \max_i \sum_{v_j \in A_i} f(\{v_j\}) + \pi(R_L) + \sum_{v_j \in R \setminus R_L} \pi(\{v_j\}) \\
\leq \max_i |A_i|a^* + r^* + |R \setminus R_L|r^* \\
\leq \left( \left\lceil \frac{n}{m} \right\rceil + 1 \right)(a^* + r^*),
\]

where the first inequality follows from the submodularity of \( f(\cdot) \) and the second inequality follows from \( f(\{v_j\}) \leq a^* \) for \( v_j \in V \setminus R_L \) and \( \pi(\{v_j\}) \leq r^* \) for \( v_j \in V \setminus A_L \). \( \square \)

Using the \( m \)-approximation algorithm, we can easily find lower and upper bounds on ‘a’ and ‘r’. Applying binary search in this interval, we can find the \( a = a^* \) and \( r = r^* \) in time polynomial time. In other words, we can design a \( \left\lceil \frac{n}{m} \right\rceil + 1 \)-approximation combinatorial algorithm using Algorithm 3.

Combining the above two approximation algorithms, select the minimum objective value the partition between two output partitions produced by Algorithms 2 and 3, we have

**Theorem 5.** There exists a combinatorial \( \min \{m, \left\lceil \frac{n}{m} \right\rceil + 1\} = O(\sqrt{n}) \)-approximation algorithm for the SLB with submodular penalties under the homogeneous setting.

**5. Conclusions and Future Work**

In this paper, we consider the submodular load balancing (SLB) with submodular penalties under two settings: Heterogenous setting and homogenous setting. We design a Lovász rounding algorithm achieving a worst-case guarantee of \( m + 1 \) under the heterogenous setting and a \( O(\sqrt{n}) \)-approximation combinatorial algorithm under the homogenous setting.

In Corollary 1, we present a low bound showing that there is no \( m \) approximation algorithm even for the homogeneous setting for the SLB with submodular penalties. Therefore, there is a gap of 1 between
Lovász rounding algorithm and this lower bound. Is it possible to either improve our algorithm or find a better low bound for the SLB with submodular penalties?

For the SLB with submodular penalties, our lower bounds show the impossibility of constant approximations to solve it. This means that to obtain better algorithms for specific applications, one has to resort to more restricted models, avoiding the full generality of arbitrary polymatroid functions. Thus, the SLB with submodular penalties under some restriction (such as curvature) is also a meaningful research direction.

**Author Contributions:** Conceptualization, X.L. and W.L.; methodology, W.L.; software, X.L. and P.X.; validation, X.L., P.X. and W.L.; formal analysis, X.L.; investigation, W.L.; resources, W.L.; data curation, X.L. and P.X.; 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, 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. Graham, R. Bounds for Certain Multiprocessing Anomalies. *Bell Syst. Tech. J.* **1966**, *45*, 1563–1581. [CrossRef]
2. Graham, R. Bounds on Multiprocessing Timing Anomalies. *SIAM J. Appl. Math.* **1969**, *17*, 416–429. [CrossRef]
3. Jansen, K.; Porkolab, L. Improved Approximation Schemes for Scheduling Unrelated Parallel Machines. *Math. Oper. Res.* **2001**, *26*, 324–338. [CrossRef]
4. Jansen, K.; Mastrolilli, M. Scheduling unrelated parallel machines: Linear programming strikes back. *Christ. Albrechts Univ. Kiel* **2010**.
5. Hochbaum, D.; Shmoys, D. Using dual approximation algorithms for scheduling problems: Theoretical and practical results. *J. ACM* **1987**, *34*, 144–162. [CrossRef]
6. Alon, N.; Azar, Y.; Woeginger, G.J.; Yadid, T. Approximation schemes for scheduling on parallel machines. *J. Sched.* **1998**, *1*, 56–66. [CrossRef]
7. Jansen, K. An EPTAS for Scheduling Jobs on Uniform Processors: Using an MILP Relaxation with a Constant Number of Integral Variables. *SIAM J. Discrete Math.* **2010**, *24*, 457–485. [CrossRef]
8. Jansen, K.; Klein, K.M.; Verschae, J. Closing the Gap for Makespan Scheduling via Sparsification Techniques. *Math. Oper. Res.* **2016**. [CrossRef]
9. Horowitz, E.; Sahni, S. Exact and Approximate Algorithms for Scheduling Nonidentical Processors. *J. ACM* **1976**, *23*, 317–327. [CrossRef]
10. Lenstra, J.K.; Shmoys, D.B.; Tardos, E. Approximation algorithms for scheduling unrelated parallel machines. *Math. Program.* **1987**, *46*, 259–271. [CrossRef]
11. Kones, I.; Levin, A. A Unified Framework for Designing EPTAS for Load Balancing on Parallel Machines. *Algorithmica* **2019**, *81*, 1–22. [CrossRef]
12. Bhaskara, A.; Krishnaswamy, R.; Talwar, K.; Wieder, U. Minimum Makespan Scheduling with Low Rank Processing Times. In Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 6–8 January 2013.
13. Chen, L.; Marx, D.; Ye, D.; Zhang, G. Parameterized and approximation results for scheduling with a low rank processing time matrix. In Proceedings of the 34th Symposium on Theoretical Aspects of Computer Science, Hannover, Germany, 8–11 March 2017.
14. Bartal, Y.; Leonardi, S.; Marchetti-Spaccamela, A.; Sgall, J.; Stougie, L. Multiprocessor Scheduling with Rejection. *SIAM J. Discret. Math.* **2000**, *13*, 64–78. [CrossRef]
15. Ou, J.; Zhong, X.; Wang, G. An improved heuristic for parallel machine scheduling with rejection. *Eur. J. Oper. Res.* **2015**, *241*, 653–661. [CrossRef]
16. Li, W.; Li, J.; Zhang, X.; Chen, Z. Penalty cost constrained identical parallel machine scheduling problem. *Theor. Comput. Sci.* 2016, 607, 181–192. doi:10.1016/j.tcs.2015.10.007. [CrossRef]

17. Zhong, X.; Ou, J. Improved approximation algorithms for parallel machine scheduling with release dates and job rejection. *Q. J. Oper. Res.* 2017, 15, 387–406. [CrossRef]

18. Zuo, L.; Sun, Z.; Lu, L.; Zhang, L. Single-Machine Scheduling with Rejection and an Operator Non-Availability Interval. *Mathematics* 2019, 7, 668, doi:10.3390/math7080668. [CrossRef]

19. Badanidiyuru, A.; Mirzasoleiman, B.; Karbasi, A.; Krause, A. Streaming Submodular Maximization: Massive Data Summarization on the Fly. In Proceedings of the ACM Conference on Knowledge Discovery in Databases (KDD), New York, NY, USA, 24–27 August 2014; pp. 671–680.

20. Soma, T.; Yoshida, Y. Maximizing monotone submodular functions over the integer lattice. *Math. Program.* 2018, 172, 539–563. [CrossRef]

21. Mirzasoleiman, B.; Jegelka, S.; Krause, A. Streaming Non-monotone Submodular Maximization: Personalized Video Summarization on the Fly. *arXiv* 2017, arXiv:1706.03583.

22. Wei, K.; Iyer, R.; Wang, S.; Bai, W.; Bilmes, J. Mixed Robust/Average Submodular Partitioning: Fast Algorithms, Guarantees, and Applications. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2015; pp. 2233–2241.

23. Svitkina, Z.; Fleischer, L. Submodular Approximation: Sampling-based Algorithms and Lower Bounds. *SIAM J. Comput.* 2011, 40, 1715–1737. [CrossRef]

24. Goemans, M.; Harvey, N.; Iwata, S.; Mirrokni, V. Approximating Submodular Functions Everywhere. In Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, New York, NY, USA, 4–6 January 2009; pp. 535–544.

25. 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]

26. Liu, X.; Li, W. Combinatorial approximation algorithms for the submodular multicut problem in trees with submodular penalties. *J. Comb. Optim.* 2018, 31, 318–330. [CrossRef]

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

© 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/).