Algorithmica (2018) 80:2957–2971
https://doi.org/10.1007/s00453-017-0363-8

A Note on Submodular Function Minimization with Covering Type Linear Constraints

Naoyuki Kamiyama[1,2]

Received: 19 January 2017 / Accepted: 10 August 2017 / Published online: 14 August 2017

Abstract In this paper, we consider the non-negative submodular function minimization problem with covering type linear constraints. Assume that there exist m linear constraints, and we denote by Δ_i the number of non-zero coefficients in the ith constraints. Furthermore, we assume that $\Delta_1 \geq \Delta_2 \geq \cdots \geq \Delta_m$. For this problem, Koufogiannakis and Young proposed a polynomial-time Δ_1-approximation algorithm. In this paper, we propose a new polynomial-time primal-dual approximation algorithm based on the approximation algorithm of Takazawa and Mizuno for the covering integer program with $\{0, 1\}$-variables and the approximation algorithm of Iwata and Nagano for the submodular function minimization problem with set covering constraints. The approximation ratio of our algorithm is $\max\{\Delta_2, \min\{\Delta_1, 1 + \Pi\}\}$, where Π is the maximum size of a connected component of the input submodular function.

Keywords Submodular function minimization · Primal-dual approximation algorithm

1 Introduction

Assume that we are given a finite set U. Then a function $f : 2^U \to \mathbb{R}$ is said to be *submodular*, if

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$$

for every pair of subsets X, Y of U, where $\mathbb{R}$ is the set of real numbers. Submodular functions play an important role in many fields, e.g., combinatorial optimization,

✉ Naoyuki Kamiyama
kamiyama@imi.kyushu-u.ac.jp

1 Institute of Mathematics for Industry, Kyushu University, Fukuoka, Japan

2 JST, PRESTO, Kawaguchi, Japan

machine learning, and game theory. One of the most fundamental problems related to submodular functions is the submodular function minimization problem. In this problem, we are given a submodular function $f : 2^U \to \mathbb{R}$, and the goal is to find a subset X of U minimizing $f(X)$ among all subsets of U, i.e., to find a minimizer of f. It is known [5,6,8,20] that this problem can be solved in polynomial time (we assume the oracle model).

In this paper, we consider constrained variants of the submodular function minimization problem. Constrained variants of the submodular function minimization problem have been extensively studied in various fields [4,7,9–15,21,23]. For example, Iwata and Nagano [9] considered the submodular function minimization problem with vertex covering constraints, set covering constraints, and edge covering constraints, and gave approximability and inapproximability. Goel, Karande, Tripathi, and Wang [4] considered the vertex cover problem, the shortest path problem, the perfect matching problem, and the minimum spanning tree problem with a monotone submodular cost function. Svitkina and Fleischer [21] also considered several optimization problems with a submodular cost function. Especially, Svitkina and Fleischer [21] proved that for the submodular function minimization problem with cardinality lower bound, there does not exist a polynomial-time $o(\sqrt{n/\ln n})$-approximation algorithm. Iyer and Bilmes [10] and Kamiyama [14] considered the submodular function minimization problem with submodular set covering constraints. Furthermore, Jegelka and Bilmes [13] considered the submodular function minimization problem with cut constraints. Koufogiannakis and Young [15] considered the monotone submodular function minimization problem with general covering constraints. Hochbaum [7] considered the submodular minimization problem with linear constraints having at most two variables per inequality. Zhang and Vorobeychik [23] considered the submodular function minimization problem with routing constraints.

In this paper, we consider the non-negative submodular function minimization problem with covering type linear constraints. Assume that there exist m linear constraints, and we denote by Δ_i the number of non-zero coefficients in the ith constraints. Furthermore, we assume that $\Delta_1 \geq \Delta_2 \geq \cdots \geq \Delta_m$. For this problem, Koufogiannakis and Young [15] proposed a polynomial-time Δ_1-approximation algorithm. In this paper, we propose a new polynomial-time primal-dual approximation algorithm based on the approximation algorithm of Takazawa and Mizuno [22] for the covering integer program with {0, 1}-variables and the approximation algorithm of Iwata and Nagano [9] for the submodular function minimization problem with set covering constraints. The approximation ratio of our algorithm is

$$\max\{\Delta_2, \min\{\Delta_1, 1 + \Pi\}\},$$

where Π is the maximum size of a connected component of the input submodular function (see the next section for its formal definition). It is not difficult to see that the approximation ratio of our algorithm is at most Δ_1. Furthermore, if Π is small (i.e., the input submodular function is close to a linear function) and Δ_2 is also small, then our approximation can improve the algorithm of Koufogiannakis and Young [15]. For example, in the minimum knapsack problem with a forcing graph (see, e.g., [22] for its formal definition), Δ_1 is large, but Δ_2 is small.

2 Preliminaries

We denote by $\mathbb{R}$ and $\mathbb{R}_+$ the sets of real numbers and non-negative real numbers, respectively. For each finite set U, each vector v in $\mathbb{R}^U$, and each subset X of U, we define $v(X) := \sum_{u \in X} v(u)$.

Throughout this paper, we are given finite sets N and $M = \{1, 2, \ldots, m\}$ such that $m \geq 2$, and a non-negative submodular function $\rho \colon 2^N \to \mathbb{R}_+$ such that $\rho(\emptyset) = 0$. We assume that for every subset X of N, we can compute $\rho(X)$ in time bounded by a polynomial in $|N|$. Furthermore, we are given vectors a in $\mathbb{R}_+^{M \times N}$ and b in $\mathbb{R}_+^M$. For each subset X of N, we define the vector χ_X in $\{0, 1\}^N$ by

$$\chi_X(j) := \begin{cases} 1 & \text{if } j \in X \\ 0 & \text{if } j \in N \setminus X. \end{cases}$$

Then we consider the following problem SCIP.

$$\begin{array}{ll} \text{Minimize} & \rho(X) \\ \text{subject to} & \displaystyle\sum_{j \in N} a(i, j) \chi_X(j) \geq b(i) \quad (i \in M) \\ & X \subseteq N. \end{array}$$

Without loss of generality, we assume that for every element i in M,

$$\sum_{j \in N} a(i, j) \geq b(i). \tag{1}$$

Otherwise, there does not exist a feasible solution of SCIP.

For each element i in M, we define Δ_i as the number of elements j in N such that $a(i, j) \neq 0$. Without loss of generality, we assume that $\Delta_1 \geq \Delta_2 \geq \cdots \geq \Delta_m$.

A subset X of N is said to be *separable*, if there exists a non-empty proper subset Y of X such that

$$\rho(X) = \rho(Y) + \rho(X \setminus Y).$$

Furthermore, a subset X of N is said to be *inseparable*, if X is not separable. It is known [1, Proposition 4.4] that N can be uniquely partitioned into non-empty subsets $I_1, I_2, \ldots, I_\delta$ satisfying the following conditions in polynomial time by using the algorithm of Queyranne [19]. (For completeness, we give an algorithm for computing $I_1, I_2, \ldots, I_\delta$ in Appendix 4.)

1. I_p is inseparable for every integer p in $\{1, 2, \ldots, \delta\}$.
2. For every subset X of N,

$$\rho(X) = \rho(X \cap I_1) + \rho(X \cap I_2) + \cdots + \rho(X \cap I_\delta).$$

Define

$$\Pi := \max\{|I_1|, |I_2|, \ldots, |I_\delta|\},$$

and we call Π the *dependency* of ρ. In this paper, we propose a polynomial-time approximation algorithm for SCIP whose approximation ratio is

$$\max\{\Delta_2, \min\{\Delta_1, 1+\Pi\}\}.$$

For SCIP, Koufogiannakis and Young [15] proved that if ρ is monotone, i.e., $\rho(X) \le \rho(Y)$ for every pair of subsets X, Y of N such that $X \subseteq Y$, then there exists a Δ_1-approximation algorithm. (See [9, p.675] for the monotonicity of an objective function.) Iwata and Nagano [9] considered the case where $a(i, j) \in \{0, 1\}$ and $b(i) = 1$ for every element i in M and every element j in N, and proposed a Δ_1-approximation algorithm. Notice that if there exists a vector c in $\mathbb{R}_+^N$ such that $\rho(X) = c(X)$ holds for every subset X of N, then the dependency Π is equal to 1. Thus, if we assume that $\Delta_2 \ge 2$, then the approximation ratio of our algorithm is Δ_2. This implies that our result can be regarded as a generalization of the Δ_2-approximation algorithm of Takazawa and Mizuno [22] for the covering integer program with $\{0, 1\}$-variables.

3 Algorithm

For proposing an approximation algorithm for SCIP, we need to introduce a linear programming relaxation of SCIP. This approach was proposed by Iwata and Nagano [9] for the submodular function minimization problem with set covering constraints.

We first define the function $\widehat{\rho}\colon \mathbb{R}_+^N \to \mathbb{R}_+$ called the *Lovász extension* of ρ [16]. Assume that we are given a vector v in $\mathbb{R}_+^N$. Furthermore, we assume that for non-negative real numbers $\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_s$ such that $\hat{v}_1 > \hat{v}_2 > \cdots > \hat{v}_s$, we have $\{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_s\} = \{v(j) \mid j \in N\}$. Then for each integer p in $\{1, 2, \ldots, s\}$, we define N_p by

$$N_p := \{j \in N \mid v(j) \ge \hat{v}_p\}.$$

Then we define $\widehat{\rho}(v)$ by

$$\widehat{\rho}(v) := \sum_{p=1}^{s} \left(\hat{v}_p - \hat{v}_{p+1}\right) \rho(N_p),$$

where we define $\hat{v}_{s+1} := 0$. It is known [3] that

$$\widehat{\rho}(v) = \max_{z \in \mathrm{P}(\rho)} \sum_{j \in N} v(j) z(j), \tag{2}$$

where we define $\mathrm{P}(\rho)$ by

$$\mathrm{P}(\rho) := \{z \in \mathbb{R}^N \mid z(X) \le \rho(X) \text{ for every subset } X \text{ of } N\}.$$

By considering the dual problem of (2), we can see that for every vector v in $\mathbb{R}_+^N$, $\widehat{\rho}(v)$ is equal to the optimal objective value of the following problem (see, e.g., [9]).

$$\begin{array}{ll} \text{Minimize} & \sum_{X \subseteq N} \rho(X)\xi(X) \\ \text{subject to} & \sum_{X \subseteq N:\, j \in X} \xi(X) = v(j) \quad (j \in N) \\ & \xi \in \mathbb{R}_+^{2^N}. \end{array} \tag{3}$$

It is not difficult to see that for every subset X of N, $\rho(X) = \widehat{\rho}(\chi_X)$. Thus, SCIP is equivalent to the following problem.

$$\begin{array}{ll} \text{Minimize} & \widehat{\rho}(x) \\ \text{subject to} & \sum_{j \in N} a(i, j)x(j) \geq b(i) \quad (i \in M) \\ & x \in \{0, 1\}^N. \end{array} \tag{4}$$

Define the vectors $\overline{a}$ in $\mathbb{R}_+^{M \times N \times 2^N}$ and $\overline{b}$ in $\mathbb{R}_+^{M \times 2^N}$ by

$$\overline{b}(i, A) := \max \Big\{0, b(i) - \sum_{j \in A} a(i, j)\Big\},$$
$$\overline{a}(i, j, A) := \min\{a(i, j), \overline{b}(i, A)\}.$$

Then we consider the following problem.

$$\begin{array}{ll} \text{Minimize} & \widehat{\rho}(x) \\ \text{subject to} & \sum_{j \in N \setminus A} \overline{a}(i, j, A)x(j) \geq \overline{b}(i, A) \quad (i \in M, A \subseteq N) \\ & x \in \{0, 1\}^N. \end{array} \tag{5}$$

The constraints of (5) are based on the results of [1,2]. It is known [1,2] that for every vector x in $\{0, 1\}^N$, x is a feasible solution of the problem (4) if and only if x is a feasible solution of the problem (5). We give the proof of this statement for completeness.

Theorem 1 *For every vector x in $\{0, 1\}^N$, x is a feasible solution of the problem* (4) *if and only if x is a feasible solution of the problem* (5)*.*

Proof Let us fix a vector x in $\{0, 1\}^N$ and an element i in M. Assume that x is a feasible solution of the problem (4). Let A be a subset of N. If there exists an element j^* in $N \setminus A$ such that $x(j^*) = 1$ and $a(i, j^*) \geq \overline{b}(i, A)$, then since $\overline{a}(i, j, A) \geq 0$ for every element j in N,

$$\sum_{j \in N \setminus A} \overline{a}(i, j, A)x(j) \geq \overline{a}(i, j^*, A) = \overline{b}(i, A).$$

Assume that $a(i, j) < \overline{b}(i, A)$ for every element j in $N \setminus A$ such that $x(j) = 1$. Since $\overline{a}(i, j, A) \geq 0$ for every element j in N,

$$\sum_{j \in N \setminus A} \overline{a}(i, j, A)x(j) \geq 0.$$

Furthermore, since

$$\sum_{j \in N} a(i, j)x(j) \geq b(i),$$

we have

$$\begin{aligned}\sum_{j \in N \setminus A} \overline{a}(i, j, A)x(j) &= \sum_{j \in N \setminus A} a(i, j)x(j) \\ &\geq b(i) - \sum_{j \in A} a(i, j)x(j) \\ &\geq b(i) - \sum_{j \in A} a(i, j).\end{aligned}$$

This implies that x is a feasible solution of the problem (5).

Assume that x is a feasible solution of the problem (5). Then we have

$$\sum_{j \in N} a(i, j)x(j) \geq \sum_{j \in N} \overline{a}(i, j, \emptyset)x(j) \geq \overline{b}(i, \emptyset) \geq b(i).$$

This implies that x is a feasible solution of the problem (4). □

We consider the following relaxation problem RP of the problem (5). Notice that Theorem 1 implies that RP is a relaxation problem of the problem (4).

$$\begin{aligned}&\text{Minimize} && \widehat{\rho}(x) \\ &\text{subject to} && \sum_{j \in N \setminus A} \overline{a}(i, j, A)x(j) \geq \overline{b}(i, A) \quad (i \in M, A \subseteq N) \\ &&& x \in \mathbb{R}_{+}^{N}.\end{aligned}$$

Since for every vector v in $\mathbb{R}_{+}^{N}$, $\widehat{\rho}(v)$ is equal to the optimal objective value of the problem (3), the optimal objective value of RP is equal to that of the following problem LP.

$$\begin{aligned}&\text{Minimize} && \sum_{X \subseteq N} \rho(X)\xi(X) \\ &\text{subject to} && \sum_{j \in N \setminus A} \overline{a}(i, j, A)x(j) \geq \overline{b}(i, A) \quad (i \in M, A \subseteq N) \\ &&& \sum_{X \subseteq N:\, j \in X} \xi(X) = x(j) \quad (j \in N) \\ &&& (x, \xi) \in \mathbb{R}^{N} \times \mathbb{R}_{+}^{2^N}.\end{aligned}$$

Notice that we neglect the redundant non-negativity constraint of x. Then the dual problem of LP can be described as follows.

$$\begin{aligned}
\text{Maximize} \quad & \sum_{i \in M} \sum_{A \subseteq N} \overline{b}(i, A) y(i, A) \\
\text{subject to} \quad & \sum_{i \in M} \sum_{A \subseteq N:\, j \notin A} \overline{a}(i, j, A) y(i, A) = z(j) \quad (j \in N) \\
& (y, z) \in \mathbb{R}_+^{M \times 2^N} \times \mathrm{P}(\rho).
\end{aligned}$$

We call this problem DLP.

Let z be a vector in $\mathrm{P}(\rho)$. Define the function $\rho - z \colon 2^N \to \mathbb{R}_+$ by $(\rho - z)(X) := \rho(X) - z(X)$. Then $\rho - z$ is submodular, and $\min_{X \subseteq N}(\rho - z)(X) = (\rho - z)(\emptyset) = 0$. Furthermore, it is not difficult to see that for every pair of minimizers X, Y of $\rho - z$, $X \cup Y$ is a minimizer of $\rho - z$. Thus, there exists the unique maximal subset X of N such that $\rho(X) = z(X)$.

We are now ready to propose our algorithm, called **Algorithm 1**. This algorithm is based on the approximation algorithm of Takazawa and Mizuno [22] for the covering integer program with $\{0, 1\}$-variables. For each element i in M and each subset S of N, we define a vector $g_{i,S}$ in $\mathbb{R}_+^N$ by

$$g_{i,S}(j) := \begin{cases} \overline{a}(i, j, S) & \text{if } j \in N \setminus S \\ 0 & \text{if } j \in S. \end{cases}$$

Then **Algorithm 1** can be described as follows. Notice that $y_1, y_2, \ldots, y_T$ are needed only for the analysis of **Algorithm 1**.

The following lemmas imply that **Algorithm 1** is well-defined and halts in finite time.

Lemma 1 *Assume that we are given an element i in M and a subset S of N such that $\overline{b}(i, S) > 0$. Then there exists an element j in $N \setminus S$ such that $\overline{a}(i, j, S) > 0$. Furthermore, there exists a subset X of N such that $g_{i,S}(X) \neq 0$.*

Proof The second statement follows from the first statement. Assume that for every element j in $N \setminus S$, $\overline{a}(i, j, S) = 0$ (notice that $\overline{a}(i, j, S) \geq 0$). Then for every element j in $N \setminus S$, since $\overline{b}(i, S) > 0$, the definition of $\overline{a}(i, j, S)$ implies that $a(i, j) = 0$. Thus, we have

$$b(i) > \sum_{j \in S} a(i, j) = \sum_{j \in N} a(i, j),$$

where the strict inequality follows from the fact that $\overline{b}(i, S) > 0$. This contradicts (1). □

Lemma 2 *Assume that we are given an element i in M, a subset S of N, and a vector z in $\mathrm{P}(\rho)$ such that $\overline{b}(i, S) > 0$. Furthermore, we assume that S is the unique maximal subset of N such that $\rho(S) = z(S)$. If we define*

$$\alpha := \min_{X \subseteq N:\, g_{i,S}(X) \neq 0} \frac{\rho(X) - z(X)}{g_{i,S}(X)}$$

Algorithm 1:

Set $t{:=}1$ and $r{:=}m$.
Define y_1, z_1 to be the zero vectors in $\mathbb{R}^{M\times 2^N}$ and $\mathbb{R}^N$, respectively.
Define S_1 to be the unique maximal subset of N such that $\rho(S_1) = z_1(S_1)$.
while $r \geq 1$ **do**
  while $\overline{b}(r, S_t) > 0$ **do**
    Define the real number α_t by

$$\alpha_t{:=} \min_{X\subseteq N:\, g_{r,S_t}(X)\neq 0} \frac{\rho(X) - z_t(X)}{g_{r,S_t}(X)}.$$

    Define the vector y_{t+1} in $\mathbb{R}^{M\times 2^N}$ by

$$y_{t+1}(i, A){:=} \begin{cases} y_t(i, A) + \alpha_t & \text{if } i = r \text{ and } A = S_t \\ y_t(i, A) & \text{otherwise.} \end{cases}$$

    Define $z_{t+1}{:=}z_t + \alpha_t \cdot g_{r,S_t}$.
    Define S_{t+1} to be the unique maximal subset of N such that $\rho(S_{t+1}) = z_{t+1}(S_{t+1})$.
    Set $t{:=}t + 1$.
  end
  Define $t(r){:=}t$, and set $r{:=}r - 1$.
end
Define $T{:=}t$.
if $t(1) = t(2)$ **then**
  Define $Q{:=}S_T$.
else
  For each integer ℓ in $\{1, 2, \dots, \delta\}$, we define

$$Q_\ell{:=}S_{T-1} \cup ((S_T \setminus S_{T-1}) \cap (I_1 \cup I_2 \cup \cdots \cup I_\ell)).$$

  Define $\beta{:=}\min\{\ell \in \{1, 2, \dots, \delta\} \mid \overline{b}(1, Q_\ell) = 0\}$ and $Q{:=}Q_\beta$.
end
Output Q, and halt.

and $z'{:=}z + \alpha \cdot g_{i,S}$, then we have

(1) $z' \in \mathrm{P}(\rho)$.

Furthermore, we define S' as the maximal subset of N such that $\rho(S') = z'(S')$. Then we have

(2) $S \subsetneq S'$.

Proof We first prove (1). For every subset X of N such that $g_{i,S}(X) = 0$, we have $z'(X) = z(X) \leq \rho(X)$. Furthermore, for every subset X of N such that $g_{i,S}(X) \neq 0$,

$$z'(X) = z(X) + \alpha \cdot g_{i,S}(X) \leq z(X) + \frac{\rho(X) - z(X)}{g_{i,S}(X)} \cdot g_{i,S}(X) = \rho(X).$$

This completes the proof.

Next we prove (2). Since $z'(j) = z(j)$ for every element j in S, $\rho(S) = z'(S)$. The maximality of S' implies that $S \subseteq S'$. Let Z be a subset of N such that $g_{i,S}(Z) \neq 0$ and

$$\alpha = \frac{\rho(Z) - z(Z)}{g_{i,S}(Z)}.$$

Then $\rho(Z) = z'(Z)$. The maximality of S' implies that $Z \subseteq S'$ holds. Furthermore, since $g_{i,S}(Z) \neq 0$, we have $Z \not\subseteq S$, which implies that $S \subsetneq S'$. This completes the proof. □

Notice that since $Q_\delta = S_T$ and $\overline{b}(1, S_T) = 0$, β is well-defined.

4 Analysis

In this section, we analyze properties of **Algorithm 1**.

We first prove that **Algorithm 1** is a polynomial-time algorithm. It follows from Lemma 2(2) that T is at most $|N| + 1$. It is known [18] that α_t can be computed in polynomial time. Furthermore, it is known (see, e.g., [17, Note 10.11]) that we can find the unique maximal subset S_{t+1} of N such that $\rho(S_{t+1}) = z_{t+1}(S_{t+1})$ in polynomial time. These imply that **Algorithm 1** is a polynomial-time algorithm.

Next we evaluate the approximation ratio.

Lemma 3 *For every integer t in $\{1, 2, \ldots, T\}$, (y_t, z_t) is a feasible solution of* DLP.

Proof We prove this lemma by induction on t. If $t = 1$, then this lemma follows from the fact that $\rho(X) \geq 0$ for every subset X of N. Assume that this lemma holds when $t = k$ (≥ 1), and then we consider the case of $t = k + 1$. Assume that $t(r + 1) < k + 1 \leq t(r)$ for an integer r in $\{1, 2, \ldots, m\}$, where we define $t(m + 1):=0$. Since $\alpha_k \geq 0$ follows from $z_k \in \mathrm{P}(\rho)$, we have $y_{k+1} \in \mathbb{R}_+^{M \times 2^N}$. Furthermore, Lemma 2(1) implies that $z_{k+1} \in \mathrm{P}(\rho)$. For every element j in S_k, $z_{k+1}(j) = z_k(j)$ and

$$\sum_{i \in M} \sum_{A \subseteq N\colon j \notin A} \overline{a}(i, j, A) y_{k+1}(i, A) = \sum_{i \in M} \sum_{A \subseteq N\colon j \notin A} \overline{a}(i, j, A) y_k(i, A).$$

For every element j in $N \setminus S_k$, $z_{k+1}(j) - z_k(j) = \overline{a}(r, j, S_k) \cdot \alpha_k$ and

$$\begin{aligned}
&\sum_{i \in M} \sum_{A \subseteq N\colon j \notin A} \overline{a}(i, j, A) y_{k+1}(i, A) - \sum_{i \in M} \sum_{A \subseteq N\colon j \notin A} \overline{a}(i, j, A) y_k(i, A) \\
&\quad = \overline{a}(r, j, S_k) \cdot (y_{k+1}(r, S_k) - y_k(r, S_k)) \\
&\quad = \overline{a}(r, j, S_k) \cdot \alpha_k.
\end{aligned}$$

This completes the proof. □

Lemma 4 *The vector χ_Q is a feasible solution of the problem (4), i.e., Q is a feasible solution of* SCIP.

Proof Let i be an element in M. Define a subset X of N by

$$X := \begin{cases} S_{t(i)} & \text{if } i \neq 1 \\ Q & \text{if } i = 1. \end{cases}$$

Since $\overline{b}(i, X) = 0$, we have

$$b(i) \leq \sum_{j \in X} a(i, j).$$

Thus, since $a(i, j) \geq 0$ for every element j in M and $X \subseteq Q$, we have

$$b(i) \leq \sum_{j \in X} a(i, j) \leq \sum_{j \in Q} a(i, j).$$

This implies that χ_Q is a feasible solution of the problem (4). This completes the proof. □

Lemma 5 *We have $\rho(Q) = z_T(Q)$.*

Proof If $t(1) = t(2)$, then $Q = S_T$, and thus this lemma follows from $z_T(S_T) = \rho(S_T)$. In what follows, we assume that $t(1) \neq t(2)$. Since $z_T \in \mathrm{P}(\rho)$,

$$z_T(S_{T-1} \cap I_p) \leq \rho(S_{T-1} \cap I_p)$$

for every integer p in $\{1, 2, \ldots, \delta\}$. Since $z_{T-1}(S_{T-1}) = \rho(S_{T-1})$ and $z_T(j) = z_{T-1}(j)$ for every element j in S_{T-1},

$$\begin{aligned} z_T(S_{T-1}) &= z_{T-1}(S_{T-1}) \\ &= \rho(S_{T-1}) \\ &= \rho(S_{T-1} \cap I_1) + \rho(S_{T-1} \cap I_2) + \cdots + \rho(S_{T-1} \cap I_\delta) \\ &\geq z_T(S_{T-1} \cap I_1) + z_T(S_{T-1} \cap I_2) + \cdots + z_T(S_{T-1} \cap I_\delta) \\ &= z_T(S_{T-1}). \end{aligned}$$

This implies that we have

$$z_T(S_{T-1} \cap I_p) = \rho(S_{T-1} \cap I_p)$$

for every integer p in $\{1, 2, \ldots, \delta\}$. In the same way, we can prove that

$$z_T(S_T \cap I_p) = \rho(S_T \cap I_p)$$

for every integer p in $\{1, 2, \ldots, \delta\}$. Thus, since

$$Q = (S_T \cap I_1) \cup \cdots \cup (S_T \cap I_\beta) \cup (S_{T-1} \cap I_{\beta+1}) \cup \cdots \cup (S_{T-1} \cap I_\delta),$$

we have

$$\begin{aligned}\rho(Q) &= \rho(S_T \cap I_1) + \cdots + \rho(S_T \cap I_\beta) + \rho(S_{T-1} \cap I_{\beta+1}) + \cdots + \rho(S_{T-1} \cap I_\delta) \\ &= z_T(S_T \cap I_1) + \cdots + z_T(S_T \cap I_\beta) + z_T(S_{T-1} \cap I_{\beta+1}) + \cdots \\ &\quad + z_T(S_{T-1} \cap I_\delta) \\ &= z_T(Q).\end{aligned}$$

This completes the proof. □

Theorem 2 **Algorithm 1** *is an approximation algorithm for* SCIP *whose approximation ratio is* $\max\{\Delta_2, \min\{\Delta_1, 1 + \Pi\}\}$.

Proof Lemma 4 implies that **Algorithm 1** is an approximation algorithm for SCIP. Let OPT be the optimal objective value of SCIP. Lemma 3 implies that

$$\sum_{i \in M} \sum_{A \subseteq N} \overline{b}(i, A) y_T(i, A) \leq \mathsf{OPT}. \tag{6}$$

Furthermore, Lemma 5 implies that

$$\begin{aligned}\rho(Q) = z_T(Q) &= \sum_{j \in Q} \sum_{i \in M} \sum_{A \subseteq N:\ j \notin A} \overline{a}(i, j, A) y_T(i, A) \\ &= \sum_{i \in M} \sum_{A \subseteq N} \sum_{j \in Q \setminus A} \overline{a}(i, j, A) y_T(i, A).\end{aligned} \tag{7}$$

Let i be an element in M. Then we have

$$\begin{aligned}\sum_{A \subseteq N} \sum_{j \in Q \setminus A} \overline{a}(i, j, A) y_T(i, A) &= \sum_{A \subseteq N} \sum_{j \in Q \setminus A:\ a(i,j) \neq 0} \overline{a}(i, j, A) y_T(i, A) \\ &\leq \sum_{A \subseteq N} \sum_{j \in Q \setminus A:\ a(i,j) \neq 0} \overline{b}(i, A) y_T(i, A) \\ &\leq \Delta_i \cdot \sum_{A \subseteq N} \overline{b}(i, A) y_T(i, A).\end{aligned} \tag{8}$$

Assume that $t(1) \neq t(2)$. Define $Q_0 := S_{T-1}$. For every subset A of S_{T-1},

$$\begin{aligned}\sum_{j \in Q_{\beta-1} \setminus A} \overline{a}(1, j, A) &\leq \sum_{j \in Q_{\beta-1} \setminus A} a(1, j) \\ &= \sum_{j \in Q_{\beta-1}} a(1, j) - \sum_{j \in A} a(1, j) \qquad (\text{by } A \subseteq Q_{\beta-1}) \\ &< b(1) - \sum_{j \in A} a(1, j) \\ &\leq \overline{b}(1, A),\end{aligned} \tag{9}$$

where the strict inequality follows from the definition of β (i.e., $\overline{b}(1, Q_{\beta-1}) > 0$). Furthermore, the definition of **Algorithm 1** and Lemma 2(2) imply that for every subset A of N, if $y_T(1, A) > 0$, then $A \subseteq S_{T-1}$. Thus,

$$\begin{aligned}&\sum_{A \subseteq N} \sum_{j \in Q \setminus A} \overline{a}(1, j, A) y_T(1, A) \\ &\quad = \sum_{A \subseteq S_{T-1}} \sum_{j \in Q \setminus A} \overline{a}(1, j, A) y_T(1, A) \\ &\quad = \sum_{A \subseteq S_{T-1}} y_T(1, A) \sum_{j \in Q \setminus A} \overline{a}(1, j, A) \\ &\quad = \sum_{A \subseteq S_{T-1}} y_T(1, A) \Bigg\{ \sum_{j \in Q_{\beta-1} \setminus A} \overline{a}(1, j, A) + \sum_{j \in Q \setminus Q_{\beta-1}} \overline{a}(1, j, A) \Bigg\} \\ &\quad \leq \sum_{A \subseteq S_{T-1}} y_T(1, A) \Bigg\{ \overline{b}(1, A) + \sum_{j \in Q \setminus Q_{\beta-1}} \overline{b}(1, A) \Bigg\} \qquad (\text{by } (9)) \\ &\quad \leq \sum_{A \subseteq S_{T-1}} y_T(1, A) \Big\{ \overline{b}(1, A) + \Pi \cdot \overline{b}(1, A) \Big\} \qquad (\text{by } |I_\beta| \leq \Pi) \\ &\quad = (1 + \Pi) \cdot \sum_{A \subseteq S_{T-1}} \overline{b}(1, A) y_T(1, A) \\ &\quad = (1 + \Pi) \cdot \sum_{A \subseteq N} \overline{b}(1, A) y_T(1, A).\end{aligned} \tag{10}$$

Notice that if $t(1) = t(2)$, then $y_T(1, A) = 0$ for every subset A of N. Thus, (6), (7), (8), and (10) imply that

$$\begin{aligned}\rho(Q) &= \sum_{i \in M} \sum_{A \subseteq N} \sum_{j \in Q \setminus A} \overline{a}(i, j, A) y_T(i, A) \\ &= \sum_{A \subseteq N} \sum_{j \in Q \setminus A} \overline{a}(1, j, A) y_T(1, A) + \sum_{i \in M \setminus \{1\}} \sum_{A \subseteq N} \sum_{j \in Q \setminus A} \overline{a}(i, j, A) y_T(i, A) \\ &\leq \min\{\Delta_1, 1 + \Pi\} \cdot \sum_{A \subseteq N} \overline{b}(1, A) y_T(1, A) + \Delta_2 \cdot \sum_{i \in M \setminus \{1\}} \sum_{A \subseteq N} \overline{b}(i, A) y_T(i, A) \\ &\leq \max\{\Delta_2, \min\{\Delta_1, 1 + \Pi\}\} \cdot \sum_{i \in M} \sum_{A \subseteq N} \overline{b}(i, A) y_T(i, A) \\ &\leq \max\{\Delta_2, \min\{\Delta_1, 1 + \Pi\}\} \cdot \mathsf{OPT}.\end{aligned}$$

This completes the proof. □

A Algorithm for Computing $I_1, I_2, \ldots, I_\delta$

It is known [1, Proposition 4.4] that we can compute $I_1, I_2, \ldots, I_\delta$ by greedily partitioning a separable subset in a current partition. Formally speaking, we can compute $I_1, I_2, \ldots, I_\delta$ by using **Algorithm 2**.

Algorithm 2:

Set $\mathcal{P} := \{N\}$.
if *there exists a separable member X in $\mathcal{P}$* **then**
  Find a non-empty proper subset Y of X such that $\rho(X) = \rho(Y) + \rho(X \setminus Y)$.
  Set $\mathcal{P} := (\mathcal{P} \setminus \{X\}) \cup \{X \setminus Y, Y\}$.
end
Output $\mathcal{P}$, and halt.

For proving that $I_1, I_2, \ldots, I_\delta$ can be computed in polynomial time, it suffices to prove that the following problem can be solved in polynomial time.

Input: A subset X of N.
Task: Decide whether there exists a non-empty proper subset Y of X such that $\rho(X) = \rho(Y) + \rho(X \setminus Y)$. If there exists such a subset Y, then find Y.

Define $\overline{\rho} \colon 2^X \to \mathbb{R}$ by

$$\overline{\rho}(Y) := \rho(Y) + \rho(X \setminus Y) - \rho(X).$$

Then it is not difficult to see that for every subset Y of X, we can compute $\overline{\rho}(Y)$ in time bounded by a polynomial in $|N|$. Furthermore, $\overline{\rho}(\emptyset) = \overline{\rho}(X) = 0$, $\overline{\rho}(Y) = \overline{\rho}(X \setminus Y)$

for every subset Y of X. For each pair of subsets Y, Z of X,

$$\begin{aligned}
\overline{\rho}(Y) + \overline{\rho}(Z) &= \rho(Y) + \rho(X \setminus Y) - \rho(X) + \rho(Z) + \rho(X \setminus Z) - \rho(X) \\
&\geq \rho(Y \cup Z) + \rho(Y \cap Z) + \rho(X \setminus (Y \cap Z)) \\
&\quad + \rho(X \setminus (Y \cup Z)) - 2\rho(X) \\
&= \rho(Y \cup Z) + \rho(X \setminus (Y \cup Z)) - \rho(X) \\
&\quad + \rho(Y \cap Z) + \rho(X \setminus (Y \cap Z)) - \rho(X) \\
&= \overline{\rho}(Y \cup Z) + \overline{\rho}(Y \cap Z).
\end{aligned}$$

That is, $\overline{\rho}$ is a submodular function. For every subset Y of X,

$$2\overline{\rho}(Y) = \overline{\rho}(Y) + \overline{\rho}(X \setminus Y) \geq \overline{\rho}(X) + \overline{\rho}(\emptyset) = 0.$$

Thus, there exists a non-empty proper subset Y of X such that $\rho(X) = \rho(Y) + \rho(X \setminus Y)$ if and only if there exists a minimizer Y of $\overline{\rho}$ such that $Y \neq \emptyset, X$. It is known [19] that we can find a non-empty proper subset Y of X minimizing $\overline{\rho}(Y)$ among all non-empty proper subsets of X in polynomial time. Let Y^* be a non-empty proper subset of X minimizing $\overline{\rho}(Y^*)$ among all non-empty proper subsets of X. If $\overline{\rho}(Y^*) > 0$ holds, then there does exist a non-empty proper subset Y of X such that $\rho(X) = \rho(Y) + \rho(X \setminus Y)$. Otherwise, Y^* is a solution of the above problem. This complete the proof.

Acknowledgements The author would like to thank Naonori Kakimura and the anonymous referees for helpful comments. This research was supported by JST PRESTO Grant Number JPMJPR14E1, Japan.

References

1. Bach, F.R.: Learning with submodular functions: a convex optimization perspective. Found. Trends Mach. Learn. **6**(2–3), 145–373 (2013)
2. Carnes, T., Shmoys, D.B.: Primal-dual schema for capacitated covering problems. Math. Program. **153**(2), 289–308 (2015)
3. Carr, R.D., Fleischer, L.K., Leung, V.J., Phillips, C.A.: Strengthening integrality gaps for capacitated network design and covering problems. In: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 106–115 (2000)
4. Edmonds, J.: Submodular functions, matroids, and certain polyhedra. In: Guy, R., Hanani, H., Sauer, N., Schönheim, J. editors, Combinatorial Structures and their Applications, pp. 69–87. Gordon and Breach (1970)
5. Goel, G., Karande, C., Tripathi, P., Wang, L.: Approximability of combinatorial problems with multi-agent submodular cost functions. In: Proceedings of the 50th Annual Symposium on Foundations of Computer Science, pp. 755–764 (2009)
6. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica **1**(2), 169–197 (1981)
7. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer, Berlin (1988)
8. Hochbaum, D.S.: Submodular problems–approximations and algorithms. Technical Report arXiv:1010.1945 (2010)
9. Iwata, S., Fleischer, L., Fujishige, S.: A combinatorial strongly polynomial algorithm for minimizing submodular functions. J. ACM **48**(4), 761–777 (2001)
10. Iwata, S., Nagano, K.: Submodular function minimization under covering constraints. In: Proceedings of the 50th Annual Symposium on Foundations of Computer Science, pp. 671–680 (2009)

11. Iyer, R.K., Bilmes, J.A.: Submodular optimization with submodular cover and submodular knapsack constraints. Adv. Neural Inf. Process. Syst. **26**, 2436–2444 (2013)
12. Iyer, R.K., Jegelka, S., Bilmes, J.A.: Curvature and optimal algorithms for learning and minimizing submodular functions. Adv. Neural Inf. Process. Syst. **26**, 2742–2750 (2013)
13. Iyer, R.K., Jegelka, S., Bilmes, J.A.: Monotone closure of relaxed constraints in submodular optimization: Connections between minimization and maximization. In: Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence, pp. 360–369 (2014)
14. Jegelka, S., Bilmes, J.A.: Graph cuts with interacting edge weights: examples, approximations, and algorithms. Mathematical Programming, pp. 1–42 (2016)
15. Kamiyama, N.: Submodular function minimization under a submodular set covering constraint. In: Proceedings of the 8th International Conference on Theory and Applications of Models of Computation, volume 6648 of Lecture Notes in Computer Science, pp. 133–141 (2011)
16. Koufogiannakis, C., Young, N.E.: Greedy Δ-approximation algorithm for covering with arbitrary constraints and submodular cost. Algorithmica **66**(1), 113–152 (2013)
17. Lovász, L.: Submodular functions and convexity. In: Bachem, A., Grötschel, M., Korte, B. (eds.) Mathematical Programming–The State of the Art, pp. 235–257. Springer-Verlag, (1983)
18. Murota, K.: Discrete Convex Analysis, volume 10 of SIAM monographs on discrete mathematics and applications. Society for Industrial and Applied Mathematics (2003)
19. Nagano, K.: A faster parametric submodular function minimization algorithm and applications. Technical Report METR 2007-43, The University of Tokyo (2007)
20. Queyranne, M.: Minimizing symmetric submodular functions. Math. Program. **82**(1), 3–12 (1998)
21. Schrijver, A.: A combinatorial algorithm minimizing submodular functions in strongly polynomial time. J. Comb. Theory Ser. B **80**(2), 346–355 (2000)
22. Svitkina, Z., Fleischer, L.: Submodular approximation: sampling-based algorithms and lower bounds. SIAM J. Comput. **40**(6), 1715–1737 (2011)
23. Takazawa, Y., Mizuno, S.: A 2-approximation algorithm for the minimum knapsack problem with a forcing graph. J. Op. Res. Soc. Japan **60**(1), 15–23 (2017)