Size-constrained Submodular Minimization through Minimum Norm Base

Kiyohito Nagano NAGANO@SAT.T.U-TOKYO.AC.JP
Collaborative Research Center for Innovative Mathematical Modelling, Institute of Industrial Science, University of Tokyo, Japan

Yoshinobu Kawahara KAWAHARA@AR.SANKEN.OSAKA-U.AC.JP
The Institute of Scientific and Industrial Research, Osaka University, Japan

Kazuyuki Aihara AIHARA@SAT.T.U-TOKYO.AC.JP
Institute of Industrial Science, University of Tokyo, Japan

Abstract

A number of combinatorial optimization problems in machine learning can be described as the problem of minimizing a submodular function. It is known that the unconstrained submodular minimization problem can be solved in strongly polynomial time. However, additional constraints make the problem intractable in many settings. In this paper, we discuss the submodular minimization under a size constraint, which is NP-hard, and generalizes the densest subgraph problem and the uniform graph partitioning problem. Because of NP-hardness, it is difficult to compute an optimal solution even for a prescribed size constraint. In our approach, we do not give approximation algorithms. Instead, the proposed algorithm computes optimal solutions for some of possible size constraints in polynomial time. Our algorithm utilizes the basic polyhedral theory associated with submodular functions. Additionally, we evaluate the performance of the proposed algorithm through computational experiments.

1. Introduction

In the areas of discrete optimization, machine learning, and other various fields, submodular functions are recognized as fundamental tools and interesting subjects of research. They appear in the systems of networks in a variety of ways and, at the same time, they naturally model economies of scale. Besides, a submodular function is known to be a discrete counterpart of a convex function (Lovász, 1983). The (unconstrained) submodular minimization is a fundamental unifying problem, and many combinatorial problems arising in machine learning, including clustering (Narasimhan et al., 2005; Nagano et al., 2010) and image segmentation (Stobbe & Krause, 2010), can be reduced to this problem. Fortunately, similarly to convexity, submodularity enables us to efficiently find an optimal solution to the unconstrained submodular minimization.

Suppose that we are given a finite set V of n elements, and a real-valued function $f : 2^V \to \mathbb{R}$ with $f(\emptyset) = 0$, where 2^V denotes the set of all subsets of V. Throughout of this paper, we suppose that f is *submodular*, that is, $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$ for all $S, T \subseteq V$. The unconstrained submodular minimization (USM) problem asks for a subset $S \subseteq V$ that minimizes $f(S)$. The first strongly polynomial algorithm for USM was described in (Grötschel et al., 1988), which relies on the ellipsoid method. The first combinatorial strongly polynomial algorithms for USM were developed by Iwata, Fleischer, and Fujishige (2001) and by Schrijver (2000). More recently, Orlin (2009) developed a faster strongly polynomial algorithm for USM, which runs in $\mathrm{O}(n^5\mathrm{EO} + n^6)$ time, where EO is the time for function evaluation. On the other hand, the Fujishige-Wolfe algorithm (refer to §7.1 of Fujishige's book (2005)) for USM is usually much faster in practice (Fujishige et al., 2006), although it does not have worst-time complexity bounds. The Fujishige-Wolfe algorithm is based on the problem

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

of finding the minimum norm base.

Constrained submodular function minimization problems have also been investigated in various contexts. Unfortunately, however, even a simple additional constraint can make the submodular minimization very difficult in many settings (Svitkina & Fleischer, 2008; Iwata & Nagano, 2009; Goel et al., 2009). In this paper, we discuss the following size-constrained submodular minimization (SSM): given a nonnegative integer $k \leq n$, the SSM problem asks for a subset $S \subseteq V$ with $|S| = k$ that minimizes $f(S)$. This problem is NP-hard, and generalizes the densest k-subgraph problem (Feige et al., 2001) and the uniform graph partitioning problem (Garey & Johnson, 1979), both of which are also NP-hard. Svitkina and Fleischer (2008) developed a sampling-based approximation algorithm for SSM, and they gave an information theoretic proof that there is no constant factor polynomial-time approximation algorithm for SSM. In our approach, we do not give approximation algorithms for SSM. Instead, our method finds a portion of optimal solutions. To be precise, we describe an algorithm that gives $K \subseteq \{0, 1, \ldots, n\}$ and solves the SSM problems for all $k \in K$ exactly. Furthermore, the proposed algorithm runs in polynomial time. This result contrasts sharply with the NP-hardness of the problem. We cannot know or specify the set K in advance, but the instance determines K. The practical performance of the proposed algorithm will be discussed in the section of computational experiments.

We remark that the proposed algorithm is quite simple and based on the polyhedral theory associated with submodular functions. This paper implies that once we obtain the minimum norm base, the SSM problems can be solved immediately in some sense. The minimum norm base can be computed within the same running time as the USM problem (Fleischer & Iwata, 2003; Nagano, 2007). Alternatively, the Fujishige-Wolfe algorithm (Fujishige, 2005) finds the minimum norm base much faster in practice.

This paper is organized as follows. In Section 2, we will see about the size-constrained submodular minimization and related problems. In Section 3, we provide the basic polyhedral theory, and introduce a parametrized fractional programming problem which is closely related to SSM. After that, we describe the algorithm that computes a portion optimal solutions to the SSM problems. Section 4 shows the validity of the algorithm proposed in Section 3. Finally, we evaluate the performance of the proposed algorithm through computational experiments in Section 5, and give concluding remarks in Section 6.

2. Size-constrained submodular minimization

Let $V = \{1, \ldots, n\}$ be a given set of n elements, and let $f : 2^V \to \mathbb{R}$ be a real-valued function defined on all the subset of V. Throughout of this paper, we suppose that $f(\emptyset) = 0$ and f is *submodular*, that is, $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$ for all $S, T \subseteq V$. In addition, we suppose that the function f is given by a value oracle.

For $k \in \{0, 1 \ldots, n\}$, we consider a size-constrained submodular minimization (SSM) problem

$$\min_S \{f(S) : S \subseteq V,\ |S| = k\}. \tag{1}$$

We say that $S \subseteq V$ is a k-subset if $|S| = k$. The problem (1) generalizes the following fundamental NP-hard problems.

Problem 2.1. (Densest k-subgraph problem) Let $G = (V, E)$ be a graph with vertex set $V = \{1, \ldots, n\}$ and edge set E. Given nonnegative edge weights w_e ($e \in E$) and an integer k, the densest k-subgraph problem asks for finding a k-subset $S \subseteq V$ that maximizes $I(S)$, where $I(S)$ is the sum of weights of edges in the subgraph induced by S. The set function $I : 2^V \to \mathbb{R}$ is *supermodular*, that is, $-I$ is submodular. This problem is NP-hard, and the approximation algorithm given in (Feige et al., 2001) achieves the best known approximation ratio of $O(n^a)$, where $a < 1/3$.

Problem 2.2 (Size-constrained minimum cut problem) Let $G = (V, E)$ be a graph, and let w_e be a nonnegative weight for each edge $e \in E$. A cut function $C : 2^V \to \mathbb{R}$ defined by $C(S) = \sum\{w_e : e \in E$ has one endpoint in S and one in $V - S\}$ $(S \subseteq V)$ is submodular. For an integer k, let us consider the problem of finding a k-subset $S \subseteq V$ that minimizes $C(S)$. When n is even, $k = n/2$, and the weights are uniform, the problem is known as the uniform graph partitioning problem (or, the minimum graph bisection problem), which is known to be NP-hard (Garey & Johnson, 1979). The algorithm developed by Krauthgamer & Feige (2006) achieves the approximation guarantee of $O(\log^{1.5} n)$ for the uniform graph partitioning problem.

For the problems 2.1 and 2.2, constant factor approximation algorithms are unknown. Clearly the size-constrained submodular minimization problem (1) is NP-hard because it generalizes NP hard problems. For the problem (1), Svitkina and Fleischer (2008) gives an $o(\sqrt{n/\ln n})$ lower bound for the approximability.

In Section 3, we describe an algorithm that gives

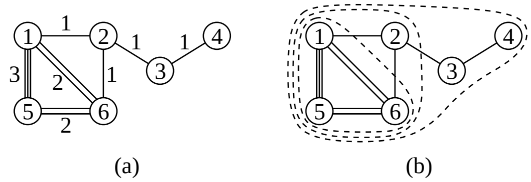


Figure 1. Finding dense subgraphs

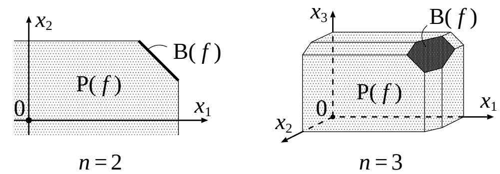


Figure 2. Polyhedra associated with submodular functions

$K \subseteq \{0, 1, \ldots, n\}$ and solves the size-constrained submodular minimization problems for all $k \in K$ exactly. Furthermore, the proposed algorithm runs in polynomial time. For example, let us consider the densest k-subgraph problems with respect to the graph with nonnegative edge weights in Figure 1 (a). Then the algorithm described in Section 3 finds the densest k-subgraphs for $k \in \{0, 3, 4, 6\}$ (see Figure 1 (b)). The proposed algorithm requires the help of the polyhedral theory associated with submodular functions.

3. The algorithm through minimum norm base

Instead of dealing with the size-constrained submodular minimization problem (1) directly, we give a parametrized fractional optimization problem, which is closely related to the problem (1). In §3.3, we will give an algorithm (Algorithm SSM), which solves the fractional optimization problem exactly and computes optimal solutions to problem (1) for some of possible size constraints. To give an efficient algorithm, the polyhedral theory of submodular functions plays an important role.

3.1. Base polyhedron and submodular minimization

Let us begin with the definitions of polyhedra associated with the submodular function f. The *submodular polyhedron* $\mathrm{P}(f) \subseteq \mathbb{R}^n$ and the *base polyhedron* $\mathrm{B}(f) \subseteq \mathbb{R}^n$ associated with the submodular function f are given by

$$\mathrm{P}(f) = \{\boldsymbol{x} \in \mathbb{R}^n : x(S) \leq f(S)\ (\forall S \subseteq V)\},$$
$$\mathrm{B}(f) = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} \in \mathrm{P}(f),\, x(V) = f(V)\},$$

where $x(S) = \sum_{i \in S} x_i$ for all $S \subseteq V$. Figure 2 illustrates these polyhedra. A point in $\mathrm{P}(f)$ is called a *subbase*, a point in $\mathrm{B}(f)$ a *base*, and an extreme point of $\mathrm{B}(f)$ an *extreme base*. For any base $x \in \mathrm{B}(f)$ and any $i \in V$ we have $f(V) - f(V \setminus \{i\}) \leq x_i \leq f(\{i\})$. Therefore, $\mathrm{B}(f)$ is bounded.

For any subbase $\boldsymbol{x}$, we say $S \subseteq V$ is $\boldsymbol{x}$*-tight* if $x(S) = f(S)$. Let $\mathcal{T}(\boldsymbol{x})$ denote the collection of all $\boldsymbol{x}$-tight subsets. The submodularity of f implies that $\mathcal{T}(\boldsymbol{x})$ is closed under union and intersection. In other words, if S and T are $\boldsymbol{x}$-tight, then $S \cup T$ and $S \cap T$ are also $\boldsymbol{x}$-tight. Thus, $\mathcal{T}(\boldsymbol{x})$ has a unique maximal (minimal) element.

Let $\boldsymbol{x}^* \in \mathrm{B}(f)$ be the minimum norm base. That is, $\boldsymbol{x}^*$ is the optimal solution to

$$\min\{\sum_{i=1}^{n} x_i^2 : \boldsymbol{x} \in \mathrm{B}(f)\}.$$

The minimum norm base and the unconstrained submodular minimization are closely related. Let $\mathrm{Arg\,min} f$ denote the collection of all minimizers of f. The submodularity of f implies that $\mathrm{Arg\,min} f$ is closed under union and intersection. Thus there exists a unique minimal minimizer as well as a unique maximal minimizer. Define

$$A_- = \{i \in V : x_i^* < 0\}, \quad A_0 = \{i \in V : x_i^* \leq 0\}.$$

Then A_- is the unique minimal minimizer of f and A_0 is the unique maximal minimizer of f (see Lemma 7.4 of Fujishige's book (2005)). Based on this fact, the Fujishige-Wolfe algorithm (Fujishige, 2005) for the unconstrained submodular minimization is constructed.

The Fujishige-Wolfe algorithm for submodular minimization just utilizes a partial information about the minimum norm base. On the contrary the proposed algorithm described in §3.3 will utilize the full information about the minimum norm base.

3.2. Fractional optimization problem

Without loss of generality, we can assume that f is nonnegative by resetting $f(S) := f(S) + \alpha|S|$ $(S \subseteq V)$ using an appropriate α. For example, consider any fixed linear ordering $L = (i_1, i_2, \ldots, i_n)$ on V, set

$$\alpha := \max\{f(L(i_j) - \{i_j\}) - f(L(i_j)) : j = 1, \ldots, n\},$$

where $L(i_j) = \{i_1, \ldots, i_j\}$ for each $j = 1, \ldots, n$, and reset $f(S) := f(S) + \alpha|S|$ for all $S \subseteq V$. Then, the new function f is submodular and nonnegative[1]. In addi-

[1]Submodularity is obvious. By the validity of the greedy algorithm of Edmonds (1970), $-\alpha \mathbf{1}_n$ is a subbase, where $\mathbf{1}_n$ is an n-dimensional all-one vector. Thus, the new function f nonnegative.

tion, this replacement does not change the structure of optimal solutions to the problem (1).

Now we introduce a parametrized fractional optimization problem. For a real parameter $\theta \in [0, n) = \{\theta' \in \mathbb{R} : 0 \le \theta' < n\}$, let us consider the following problem:

$$\lambda(\theta) := \min_S \left\{ \frac{f(S)}{|S|-\theta} : S \subseteq V,\ |S| > \theta \right\}. \qquad (2)$$

For all $\theta \in [0, n)$, let $S^{(\theta)} \subseteq V$ be an optimal solution to (2). The following lemma provides a connection between the problems (1) and (2).

Lemma 1. *For $\theta \in [0, n)$, a subset $S^{(\theta)} \subseteq V$ is an optimal solution to the size-constrained submodular minimization problem (1) with respect to $k = |S^{(\theta)}|$.*

Proof. By definition, we have $k > \theta$ and $f(S^{(\theta)})/(k-\theta) \le f(S)/(k-\theta)$ for any subset S with $|S| = k$. □

For the parametric fractional problem (2), we construct a polynomial time algorithm, or a simple algorithm which is much faster in practice. Therefore, in view of Lemma 1, we obtain an algorithm that gives $K \subseteq \{0, 1, \dots, n\}$ and solves the size-constrained submodular minimization problems for all $k \in K$ exactly. Although the problem (1) is NP-hard, it is interesting that we can partially solve the size-constrained submodular minimization problems. This is similar to the result in (Nagano et al., 2010) for submodular clustering problems.

Let us examine the structure of the optimal solutions to the fractional problem (2). For any $\theta \in [0, n)$, the optimal value $\lambda(\theta)$ of (2) is represented as

$$\begin{aligned}\lambda(\theta) &= \max\{\lambda \in \mathbb{R}_{\ge 0} : \lambda \le \tfrac{f(S)}{|S|-\theta},\ \forall S \text{ with } |S| > \theta\} \\ &= \max\{\lambda \in \mathbb{R}_{\ge 0} : -\theta\lambda \le f(S) - |S|\lambda, \\ &\qquad\qquad\qquad\qquad \forall S \text{ with } |S| > \theta\} \\ &= \max\{\lambda \in \mathbb{R}_{\ge 0} : -\theta\lambda \le f(S) - |S|\lambda,\ \forall S\},\end{aligned}$$

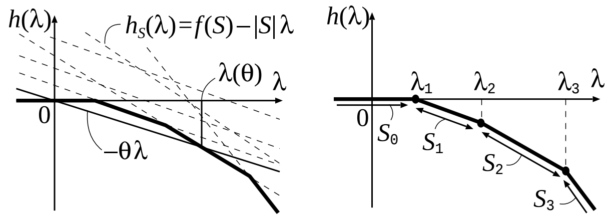
where the last equality follow from the nonnegativity of f. Define a function $h : \mathbb{R} \to \mathbb{R}$ as

$$h(\lambda) = \min_S \{f(S) - |S|\lambda : S \subseteq V\} \quad (\lambda \in \mathbb{R}). \qquad (3)$$

Then, we have

$$\lambda(\theta) = \max\{\lambda \in \mathbb{R} : -\theta\lambda \le h(\lambda)\}. \qquad (4)$$

For each subset $S \subseteq V$, define a linear function $h_S : \mathbb{R} \to \mathbb{R}$ as $h_S(\lambda) = f(S) - |S|\lambda$. Since h is the minimum of these linear functions, h is a piecewise-linear concave function. The function h is illustrated in Figure 3 (a) by the thick curve. In view of (4), the value $\lambda(\theta)$ can be obtained by solving the equation $-\theta\lambda = h(\lambda)$ (see also Figure 3 (a)). Moreover, an optimal solution to the problem (2) can be characterized as follows.



(a) (b)

Figure 3. The structure of the function h

Lemma 2. *Given a parameter $\theta \in [0, n)$, let $S \subseteq V$ be a subset such that $|S| > \theta$ and S determines h at $\lambda(\theta)$. Then S is a minimizer of the problem (2).*

Proof. Since $-\theta\lambda(\theta) = h(\lambda(\theta)) = f(S) - |S|\lambda(\theta)$, we have $\lambda(\theta) = f(S)/(|S| - \theta)$. For any subset $T \subseteq V$ with $|T| > \theta$, we have $-\theta\lambda(\theta) \le f(T) - |T|\lambda(\theta)$, and thus $\lambda(\theta) \le f(T)/(|T| - \theta)$. □

Suppose that the slopes of h take values $-s_0 > -s_1 > -s_2 > \cdots > -s_d$. Clearly, we have $s_0 = 0$, $s_d = n$ and $d \le n$. Then $\mathbb{R}$ is split into $d+1$ subintervals $R_0 = [-\infty, \lambda_1)$, $R_1 = [\lambda_1, \lambda_2)$, $\dots$, $R_d = [\lambda_d, +\infty)$ such that, for each $j = 0, \dots, d$, the function h is linear and its slope is $-s_j$ on R_j. Let $S_0, \dots, S_d$ be subsets of V such that, for each $j = 0, 1, \dots, d$, the subset S_j determines h at all $\lambda \in R_j$. Figure 3 (b) illustrates the structure of the function h.

By Lemma 2, for any $\theta \in [0, n)$, if $\lambda(\theta) \in R_j$ then S_j is an optimal solution to (2). Therefore, if we can find $S_0, \dots, S_d$ then the parametrized fractional problem (2) will be completely solved.

3.3. Algorithm

Let us see that the values $\lambda_1, \dots, \lambda_d$ and the subsets $S_0, \dots, S_d$ defined in §3.2 can be found using the minimum norm base. That is, the problem (2) can be solved with the aid of the minimum norm base.

Consider the algorithm SSM described below:

Algorithm SSM(f)

Input: A submodular function f.
Output: Subsets $T_0, T_1, \dots, T_d \subseteq V$. ($d$ is not determined in advance.)

1: Compute the minimum norm base $\boldsymbol{x}^* \in \mathrm{B}(f)$.
2: Let $\xi_1 < \xi_2 < \cdots < \xi_d$ be distinct values of $\boldsymbol{x}^*$. Return $T_0 := \emptyset$, and return $T_j := \{i \in V : x_i^* \le \xi_j\}$ for all $j = 1, \dots, d$.

Surprisingly, we have the following equalities.

Lemma 3. *Regarding the subsets $S_0, \ldots, S_d$ defined in §3.2 and the subsets $T_0, \ldots, T_d$ returned by* SSM*, we have $S_j = T_j$ for each $j = 0, \ldots, d$.*

The proof of Lemma 3 will be given in §4.1. By Lemmas 1, 2, and 3, we immediately have the main theorem.

Theorem 4. *Let $T_0, T_1, \ldots, T_d \subseteq V$ be the subsets returned by the algorithm* SSM*. For each $j \in \{0, 1, \ldots, d\}$, T_j is an optimal solution to the size-constrained submodular function problem (1) with respect to $k = |T_j|$.*

We now analyze the running time of the algorithm SSM. The main task is the computation of the minimum norm base $\boldsymbol{x}^*$. The minimum norm base $\boldsymbol{x}^*$ can be computed within the same running time as the unconstrained submodular minimization problem (Fleischer & Iwata, 2003; Nagano, 2007). Alternatively, the Fujishige-Wolfe algorithm (Fujishige, 2005) finds the minimum norm base much faster in practice. As a whole, the running time of SSM is almost the same as the unconstrained submodular minimization.

4. Analysis of the algorithm

In §4.1, we give a proof of the key lemma (Lemma 3) for the validity of the algorithm SSM. (Note that Lemma 3 can be derived from Theorem 9.5 and Corollary 9.6 in Fujishige's book (2005). In this paper, we will give a simple and direct proof of the lemma.) In §4.2, let us see some directions for use of the algorithm SSM.

4.1. Validity of the proposed algorithm

The classical result of Fujishige (1980) enables us to show Lemma 3. We describe the monotone algorithm of Fujishige (1980) (see also §9.2 of (Fujishige, 2005)), which finds the minimum norm base $\boldsymbol{x}^* \in \mathrm{B}(f)$. For all $S \subseteq V$, let $\boldsymbol{I}_S \in \{0, 1\}^n$ denote the characteristic vector of S.[2]

The monotone algorithm starts with an interior point of $\mathrm{P}(f)$ in the form of $\boldsymbol{x}^{(0)} = \delta_0 \boldsymbol{I}_V \in \mathrm{P}(f)$ and subsets $V_0 := V$, $U_0 := \emptyset$. At the beginning of the j-th iteration, the algorithm holds a subbase $\boldsymbol{x}^{(j-1)} \in \mathrm{P}(f)$, the maximal $\boldsymbol{x}^{(j-1)}$-tight subset U_{j-1} and the subset $V_{j-1} = V - U_{j-1}$. The algorithm computes $\delta_j := \max\{\delta \in \mathbb{R} : \boldsymbol{x}^{(j-1)} + \delta \boldsymbol{I}_{V_{j-1}} \in \mathrm{P}(f)\}$, and sets $\boldsymbol{x}^{(j)} := \boldsymbol{x}^{(j-1)} + \delta_j \boldsymbol{I}_{V_{j-1}}$. If $\boldsymbol{x}^{(j)}$ is a base, then the algorithm returns $\boldsymbol{x}^* := \boldsymbol{x}^{(j)}$. Otherwise, the algorithm goes to the next iteration.

[2] For example in case of $|V| = 6$, the characteristic vector of $S = \{1, 3, 6\}$ becomes $\boldsymbol{I}_S = (1, 0, 1, 0, 0, 1)$.

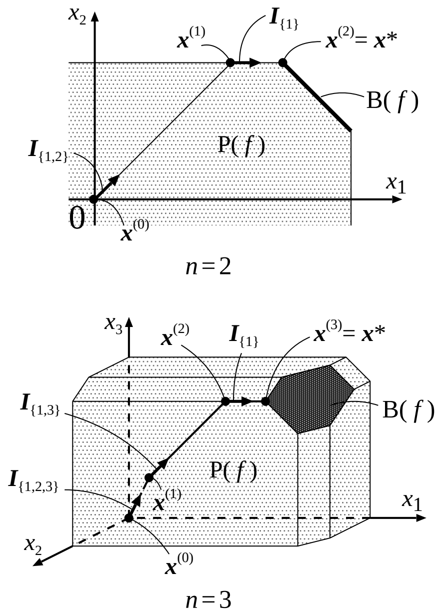


Figure 4. Algorithm MA

The monotone algorithm (Fujishige, 1980) is described more precisely as follows.

Algorithm MA(f) (Fujishige, 1980)
Input: A submodular function f.
Output: $\boldsymbol{x}^* \in \mathrm{B}(f)$.

0: Choose $\delta_0 \in \mathbb{R}$ such that $\delta_0 \boldsymbol{I}_V$ is an interior point of $\mathrm{P}(f)$. Set $\boldsymbol{x}^{(0)} := \delta_0 \boldsymbol{I}_V$. Set $V_0 := V$, $U_0 := \emptyset$, and $j := 1$.
1: Compute $\delta_j := \max\{\delta \in \mathbb{R} : \boldsymbol{x}^{(j-1)} + \delta \boldsymbol{I}_{V_{j-1}} \in \mathrm{P}(f)\}$. Set $\boldsymbol{x}^{(j)} := \boldsymbol{x}^{(j-1)} + \delta_j \boldsymbol{I}_{V_{j-1}}$. Let $U_j \subseteq V$ be the unique maximal $\boldsymbol{x}^{(j)}$-tight subset. Set $V_j := V - U_j$.
2: If $U_j = V$ ($\boldsymbol{x}^{(j)}$ is a base), then return $\boldsymbol{x}^* := \boldsymbol{x}^{(j)}$, and stop. Otherwise, set $j := j + 1$ and go to Step 1.

Figure 4 illustrates the process of the algorithm MA.

Theorem 5 (Fujishige(1980))**.** *The vector $\boldsymbol{x}^*$ returned by the algorithm* MA(f) *is the minimum norm base in* $\mathrm{B}(f)$.

It is easy to observe the following properties:

(MA-1) $\emptyset = U_0 \subsetneq \cdots \subsetneq U_d = V$, $V = V_0 \supsetneq \cdots \supsetneq V_d = \emptyset$, and $V_j = V - U_j$ for each j;

(MA-2) $\delta_j > 0$ for each $j = 1, \ldots, d$;

(MA-3) $U_0, U_1, \ldots, U_d$ are $\boldsymbol{x}^*$-tight.

Now we can give a proof of Lemma 3.

Proof of Lemma 3. In the first step, we will see that $T_j = U_j$ for each j, where each T_j is returned by the algorithm SSM. In view of the algorithm MA, the minimum norm base $\boldsymbol{x}^*$ can be represented as

$$\boldsymbol{x}^* = \delta_0 \boldsymbol{I}_V + \sum_{j=1}^{d} \delta_j \boldsymbol{I}_{V_{j-1}}. \tag{5}$$

Let us remember the definitions of the subsets $T_1, \ldots, T_d$ and the values $\xi_1, \ldots, \xi_d$ in the algorithm SSM. By (5), (MA-1) and (MA-2), we have

$$\begin{aligned} T_j &= U_j && \text{for each } j = 0, \ldots, d, \\ \xi_j &= \textstyle\sum_{j'=0}^{j} \delta_{j'} && \text{for each } j = 1, \ldots, d. \end{aligned} \tag{6}$$

In the next step, we will see that $S_j = U_j$ for each j, where each S_j is defined in §3.2. For all $\xi \in \mathbb{R}$, define a set function $g_\xi : 2^V \to \mathbb{R}$ as

$$g_\xi(S) = \textstyle\sum_{i \in S}(x_i^* - \xi) = x^*(S) - |S|\xi \quad (S \subseteq V).$$

The values $\xi_1 < \xi_2 < \cdots < \xi_d$ are distinct values of $\boldsymbol{x}^*$. For convenience, we set $\xi_0 = -\infty$ and $\xi_{d+1} = +\infty$. Fix any $j \in \{0, \ldots, d\}$, and fix any ξ with $\xi_j < \xi < \xi_{j+1}$. Clearly, U_j is the unique minimizer of g_ξ. Since U_j is $\boldsymbol{x}^*$-tight, we have

$$g_\xi(U_j) = x^*(U_j) - |U_j|\xi = f(U_j) - |U_j|\xi.$$

For any $S \subseteq V$ with $S \neq U_j$, we have

$$g_\xi(U_j) < g_\xi(S) = x^*(S) - |S|\xi \leq f(S) - |S|\xi.$$

Thus, for any $S \subseteq V$ with $S \neq U_j$, we have

$$f(U_j) - |U_j|\xi < f(S) - |S|\xi.$$

Therefore, U_j determines the function value of h at ξ. As a result, we obtain

$$\begin{aligned} S_j &= U_j && \text{for each } j = 0, \ldots, d, \\ \xi_j &= \lambda_j && \text{for each } j = 1, \ldots, d. \end{aligned} \tag{7}$$

By (6) and (7), we have $T_j = S_j$ for each $j = 0, \ldots, d$, which completes the proof of Lemma 3. □

4.2. Directions for use of the algorithm

The algorithm SSM can be applied to the densest k-subgraph problem (Problem 2.1) directly because we can expect that the minimum norm base $\boldsymbol{x}^*$ has a somewhat complicated structure (see also Figure 1 and computational experiments in Section 5). However, in the case of the size-constrained minimum cut problem (Problem 2.2), the algorithm SSM is not effective. That is why the minimum norm base $\boldsymbol{x}^*$ is always the all-zero vector. In order to overcome this shortcoming, let us consider the following alternative variant of the size-constrained minimum cut problem.

Problem 4.1 (Size-constrained minimum s-t cut problem) Let $G = (V, E)$ be a graph with nonnegative weights. Let s and t are distinct nodes in V For an integer k, let us consider the problem of minimizing the cut function value $C(S)$ among k-subset $S \subseteq V$ which includes s and excludes t.

We can apply the algorithm SSM to the size-constrained minimum s-t cut problem. Furthermore, by choosing a number of (s, t) pairs, it could be possible to obtain a good approximate solution to the size-constrained minimum cut problem.

5. Experimental results

We empirically investigated the property of the proposed algorithm using synthetic datasets in Section 5.1, and then apply the algorithm to the real-world application of bioinformatics in Section 5.2.

5.1. Artificial data

Here, we investigate the property of the proposed algorithm using synthetic datasets generated using the GENRMF generator.[3] The GENRMF generates a network with b grid-like frames of size $(a \times a)$. The number of vertices is $a^2 b$ and that of arcs $5a^2b - 4ab - a^2$. All vertices in each frame are connected to its grid neighbors and each vertex is connected by an arc to a vertex randomly chosen from the next frame. Arc capacities within a frame are $c_2 \times a \times a$ and those between frames are randomly selected integers from the range $[c_1, c_2]$. In this experiments, we set $c_1 = 1$ and $c_2 = 100$. Also, we used two-types of datasets with different ratio $(a : b)$: *Genrmf-long* ($a = 2^{x/4}$ and $b = 2^{x/2}$) and *Genrmf-wide* ($a = 2^{2x/5}$ and $b = 2^{x/5}$).

Table 1 shows the number of subsets found by the algorithm SSM, *i.e.*, d, for several sizes of graphs n. The result shown is with the dataset generated using the software with SEED= 20 (note the tendency seems to

[3] The datasets used in the first DIMACS international algorithm implementation challenge: The core experiments, 1990. The code is available from 'http://www.informatik.uni-trier.de/~naeher/Professur/research/generators/maxflow/genrmf/index.html'.

Table 1. Number of subsets found by the algorithm SSM (*i.e.*, d) for several sizes of graphs (upper: Genrmf-long and lower: Genrmf-wide).

Genrmf-long

n	cut	s-t cut	dense
63	2	4	50
126	2	3	101
256	2	4	213
525	2	4	466
1008	2	4	868

Genrmf-wide

n	cut	s-t cut	dense
75	2	4	68
147	2	4	136
324	2	3	298
576	2	4	530
1024	2	3	975

be similar for other random-number sheets). As for s-t cut, nodes s and t are set as $s = 1$ and $t = n$, respectively. First, we confirmed that cut-function always gives trivial two solutions, *i.e.*, $\emptyset$ and V, as discussed above. Although, for s-t cut and dense functions, the algorithm provide non-trivial solutions, the obtained numbers of solutions by s-t cut and dense functions were very different, which would strongly on the structure of submodular polyhedral.

Next, Figure 5 depicts solution values versus k by the proposed algorithm SSM and the backward greedy selection, *i.e.*, starting from V and dropping a subset that gives the least solution value decrease at each iteration, for Genrmf-long ($n = 1008$). Remember that, although the SSM can give solutions only for some k, the solutions are exactly optimal for the NP-hard problem (1). As can be seen in the figure, the optimality of solutions by the greedy algorithm seems to be quite low except for very large k. Like these examples, our method would be useful to see the solution performance by other approximation algorithms.

5.2. Application to real data

As an experiment with real-world datasets, we show the results using social network data *cnr-2000*.[4] This data consists of 325,557 nodes and 3,216,152 arcs. Here, due to the computational limit, we applied the proposed algorithm to the sub-network with the first 5,000 nodes in the original network, where 31,664 arcs exist. The function used here is the intensity function I, and thus the goal of this application is to find densest subgraphs in the network.

[4]See `http://law.dsi.unimi.it/webdata/cnr-2000`.

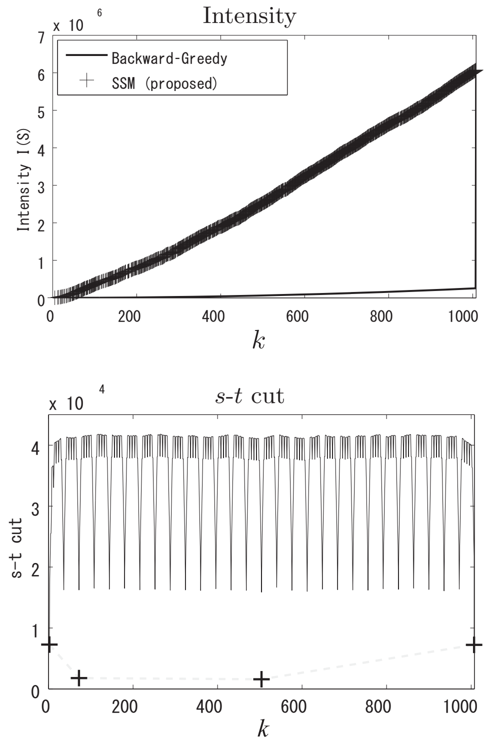


Figure 5. Intensity (upper) and s-t cut (lower) vs. k by the proposed algorithm and the backward greedy algorithm.

Our method could find exact optimal solutions for several k. Figure 6 shows the intensity $I(S)$ versus the sizes of subsets k. After increasing almost linearly to a certain k, the optimal intensity seems to be saturated.

6. Concluding remarks

The size-constrained submodular minimization is NP-hard, and it does not even have a constant factor polynomial time approximation algorithm. To this problem, we have proposed a new method that computes a portion of exact optimal solutions. Computational experiments show that our method could find several exact optimal solutions. An interesting point is that the proposed method utilizes the minimum norm base effectively. The minimum norm base can be found in polynomial time, and the Fujishige-Wolfe algorithm finds that point much faster in practice.

The Fujishige-Wolfe algorithm does not have worst-time complexity bounds, so its complexity analysis should be given in future works.

Acknowledgments

This research is partially supported by Aihara Project, the FIRST program from JSPS, JST PRESTO program, and KAKENHI (22700007 and 22700147).

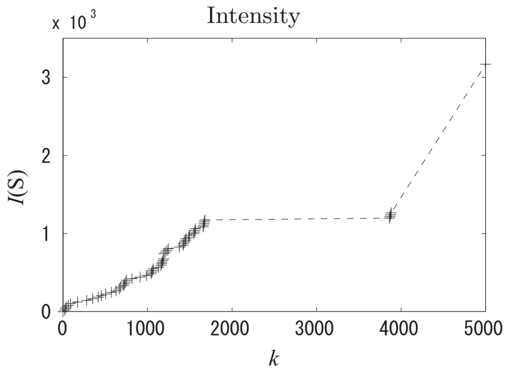


Figure 6. Optimal intensity $I(S)$ versus the size k.

References

Edmonds, J. Submodular functions, matroids, and certain polyhedra. In Guy, R., Hanani, H., Sauer, N., and Schönheim, J. (eds.), *Combinatorial Structures and Their Applications*, pp. 69–87. Gordon and Breach, 1970.

Feige, U., Kortsarz, G., and Peleg, D. The dense k-subgraph problem. *Algorithmica*, 29:410–421, 2001.

Fleischer, L. and Iwata, S. A push-relabel framework for submodular function minimization and applications to parametric optimization. *Discrete Appl. Math.*, 131:311–322, 2003.

Fujishige, S. Lexicographically optimal base of a polymatroid with respect to a weight vector. *Mathematics of Operations Research*, 5:186–196, 1980.

Fujishige, S. *Submodular Functions and Optimization*. Elsevier, 2nd edition, 2005.

Fujishige, S., Hayashi, T., and Isotani, S. The minimum-norm-point algorithm applied to submodular function minimization and linear programming. Technical report, Research Institute for Mathematical Sciences Preprint RIMS-1571, Kyoto University, Kyoto, Japan, 2006.

Garey, M. R. and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., 1979.

Goel, G., Karande, C., Tripathi, P., and Wang, L. Approximability of combinatorial problems with multi-agent submodular cost functions. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009)*, pp. 755–764, 2009.

Grötschel, M., Lovász, L., and Schrijver, A. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 2nd edition, 1988.

Iwata, S. and Nagano, K. Submodular function minimization under covering constraints. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009)*, pp. 671–680, 2009.

Iwata, S., Fleischer, L., and Fujishige, S. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48:761–777, 2001.

Krauthgamer, R. and Feige, U. A polylogarithmic approximation of the minimum bisection. *SIAM Review*, 48:99–130, 2006.

Lovász, L. Submodular functions and convexity. In Bachem, A., Grötschel, M., and Korte, B. (eds.), *Mathematical Programming — The State of the Art*, pp. 235–257. Springer-Verlag, 1983.

Nagano, K. A faster parametric submodular function minimization algorithm and applications. Technical report, METR 2007-43, University of Tokyo, Tokyo, Japan, 2007.

Nagano, K., Kawahara, Y., and Iwata, S. Minimum average cost clustering. In *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, pp. 1759–1767, 2010.

Narasimhan, M., Jojic, N., and Bilmes, J. Q-clustering. In *Advances in Neural Information Processing Systems 18 (NIPS 2005)*, pp. 979–986, 2005.

Orlin, J. B. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118:237–251, 2009.

Schrijver, A. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory (B)*, 80:346–355, 2000.

Stobbe, P. and Krause, A. Efficient minimization of decomposable submodular functions. In *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, pp. 2208–2216, 2010.

Svitkina, Z. and Fleischer, L. Submodular approximation: sampling-based algorithms and lower bounds. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, pp. 697–706, 2008.