Journal of Combinatorial Mathematics and Combinatorial Computing, 120: 191–199
DOI:10.61091/jcmcc120-016
http://www.combinatorialpress.com/jcmcc
Received 11 January 2020, Accepted 30 December 2020, Published 30 June 2024

Combinatorial Press

Article

Approximation Algorithms for the Submodular Hitting Set Problem

Shaojing Du[1,*], **Bo Hou**[1], **and Wen Liu**[1]

[1] School of Mathematical Sciences, Hebei Normal University, Shijiazhuang, 050024, P. R. China

* **Correspondence:** 15227825915@139.com

Abstract: In this paper, we study the submodular hitting set problem (SHSP), which is a variant of the hitting set problem. In the SHSP, we are given a supergraph $H = (V, C)$ and a nonnegative submodular function on the set 2^V. The objective is to determine a vertex subset to cover all hyperedges such that the cost of submodular covering is minimized. Our main work is to present a rounding algorithm and a primal-dual algorithm respectively for the SHSP and prove that they both have the approximation ratio k, where k is the maximum number of vertices in all hyperedges.

Keywords: Submodular, Hitting set, Rounding, Primal-dual, Approximation algorithm

1. Introduction

The hitting set problem involves a supergraph $H = (V, C)$, where V is a finite set with elements called *vertices* and C is a family of subsets of V called *hyperedges*. Let k be the maximum size of all hyperedges and Δ be the maximum degree of all vertices. A *hitting set* of H, also called a vertex cover of H, is a subset $X \subseteq V$ such that each hyperedge in C contains at least one vertex from X. The hitting set problem is to select a vertex subset of minimum size. For each vertex $u \in V$, if there is an associated nonnegative weight cost $w(u)$, we obtain the weighted version of the problem, the goal of which is to select a vertex subset of minimum covering cost. The minimum hitting set problem is NP-hard [1] and it is one of the fundamental combinatorial optimization problems. In this case, it is conventional that for any subset X of V the covering cost of X is the sum of the covering cost of all vertices in X, that is, $w(X) = \sum\limits_{u \in X} w(u)$. Then this function $w(\cdot)$ induces a linear function from 2^V to R_+.

Most results of decision problems in combinatorial optimization, such as trees or cuts, vertex cover in graphs, facility location, set cover, are in general asking for the linear costs. However, nonlinear costs are more realistic in many settings and there have existed some works in the area of combinatorial optimization problems with submodular costs. Assume E is a nonempty set and 2^E is the power set of E. Let $F(\cdot) : 2^E \to R_+$ be a nonnegative-valued set function. We call $F(\cdot)$ submodular if for any $X, Y \in 2^E$, $F(X \cup Y) + F(X \cap Y) \leq F(X) + F(Y)$ and $F(\cdot)$ modular if the equality holds. It is obvious that all modular functions are submodular and so all linear functions. We call $F(\cdot)$ monotone if $F(X) \leq F(Y)$ for each pair of subsets $X, Y \in 2^E$ with $X \subseteq Y$. Submodular functions occur very often in a variety of subjects such as economics, engineering, computer science and management science. There is a greedy approximation which achieves an approximation ratio of $H(\Delta)$ for the hitting set problem with linear costs, where $H(\Delta)$ is the Δ-th harmonic number [1]. Moreover, using the greedy approximation, Wan et al. [2] gave some similar results for the hitting set problem with submodular costs.

One of the most fundamental problems related to submodular functions is the submodular function minimization problem. In this problem, we are given a finite set E and a submodular function $F(\cdot)$ defined on 2^E, then the goal is to select a subset X of E minimizing $F(X)$ among the power set of E, i.e., to select a minimizer of submodular function $F(\cdot)$ over 2^E. It is known that this problem can be solved in polynomial time. Recently, constrained variants of the submodular function minimization problem have been extensively studied in various fields ([3–6]). For instance, Hochbaum [4] studied the submodular minimization problem with linear constraints having at most two variables per inequality. Kamiyama [5] studied the the submodular function minimization problem with covering type linear constraints. Koufogiannakis and Young [3] studied the monotone submodular function submodular minimization problem with general covering constrains. Iwata and Nagano [6] studied the submodular function minimization problem with set cover constraints. In this paper, we are motivated to study the submodular hitting set problem (SHSP). That is, given a nonnegative submodular function $f(\cdot) : 2^V \to R_+$, the goal of the SHSP is to select a vertex subset $X \subseteq V$ to cover all the hyperedges that minimizes the submodular cost $f(X)$. We present a rounding algorithm and a primal-dual program for the SHSP and prove that they both have the approximation ratio k.

The paper is organized as follows. In Section 2, we give some definitions about submodular function. In Section 3, we present a rounding algorithm and a primal-dual program respectively for the SHSP and prove that they both have the approximation ratio k, where k is the maximum number of vertices in all hyperedges.

2. *Lovász* Extension of a Submodular Function

In this section, we introduce the *Lovász* extension of a submodular function. Let $F(\cdot) : 2^E \to R_+$ be a submodular function.

Assume $|E| = n$. Then for any vector $\boldsymbol{z} \in R^n$ and any non-empty subset $X \subseteq E$, we define

$$\boldsymbol{z}(X) = \sum_{i \in X} \boldsymbol{z}(i),$$

where $\boldsymbol{z}(i)$ is the i-th coordinate value of vector $\boldsymbol{z}$. It is easy to see that $\boldsymbol{z}(\cdot)$ is a modular function from $2^E \to R$.

For any $X \subseteq E$, define $\boldsymbol{\chi}_X \in \{0, 1\}^n$ as follows:

$$\boldsymbol{\chi}_X(i) = \begin{cases} 1, & \text{if } i \in X, \\ 0, & \text{otherwise,} \end{cases}$$

called *the characteristic vector* of X.

Given a submodular function $F(\cdot)$, denote

$$P(F) = \{\boldsymbol{z} \mid \boldsymbol{z} \in R^n, \boldsymbol{z}(X) \le F(X), \forall X \subseteq E\}.$$

We call $P(F)$ a *submodular polyhedron* and its elements *subbases*. For any subbase $\boldsymbol{z} \in P(F)$, a subset X of E is called $\boldsymbol{z}$*-tight* whenever $\boldsymbol{z}(X) = F(X)$. There is a property for the collection of all $\boldsymbol{z}$-tight subsets.

Proposition 1. *For any given subbase $\boldsymbol{z} \in P(F)$, the collection of all $\boldsymbol{z}$-tight subsets is closed under union and intersection.*

Proof. Since $F(\cdot)$ is submodular and $\boldsymbol{z}(\cdot)$ is modular, the result holds. □

Now we introduce the *Lovász* extension of a submodular function and its properties. Given a vector $\boldsymbol{p} \in R^n_+$ and suppose $p_1, p_2, \cdots, p_l$ are all distinct coordinate values of $\boldsymbol{p}$ with $p_1 > p_2 > \cdots > p_l$. For each $j = 1, 2, \cdots, l$, denote $N_j = \{i \mid \boldsymbol{p}(i) \geq p_j\}$. Then we define a vector function over R^n_+ as follows:

$$\hat{F}(\boldsymbol{p}) = \sum_{j=1}^{l}(p_j - p_{j+1})F(N_j),$$

where $p_{l+1} = 0$. The function $\hat{F}(\cdot)$ is said to be the *Lovász* extension of $F(\cdot)$. It is easy to see that for any subset X of E, $\hat{F}(\boldsymbol{\chi}_X) = F(X)$ and $\hat{F}(\mathbf{0}) = F(\emptyset)$.

It is known from [7] that

$$\hat{F}(\boldsymbol{p}) = \max\left\{\sum_{i=1}^{n}\boldsymbol{p}(i)\boldsymbol{z}(i) \mid \boldsymbol{z} \in P(F)\right\}. \tag{1}$$

The problem (1) is a linear optimization over the submodular polyhedron, which is efficiently solved by Edmonds [8]. Combining (1) with the primal-dual theory, we get that $\hat{F}(\boldsymbol{p})$ is equal to the optimal value of the following program:

$$\begin{aligned} \min & \sum_{X\subseteq E} F(X)\cdot\zeta(X) & & \\ \text{s.t.} & \sum_{\substack{X\subseteq E\\ i\in X}}\zeta(X) = \boldsymbol{p}(i), & & \forall i \in E, \\ & \zeta(X) \geq 0, & & \forall X \subseteq E. \end{aligned} \tag{2}$$

There are some properties of the *Lovász* extension function and we list them here for later use.

Lemma 1. ([7]) A set function $F(\cdot)$ is submodular if and only if $\hat{F}(\cdot)$ is convex.

Lemma 2. ([9]) Assume $\hat{F}(\cdot)$ is the *Lovász* extension function of submodular function $F(\cdot)$, then we have

(1) (positively homogeneous) for any real number $\alpha > 0$ and any vector $\boldsymbol{v} \geq \mathbf{0}$, $\hat{F}(\alpha\boldsymbol{v}) = \alpha\hat{F}(\boldsymbol{v})$;
(2) the function $\hat{F}(\cdot)$ is a monotonically increasing convex function.

3. Approximation Algorithms

In this section, we construct a rounding algorithm and a primal-dual algorithm for the SHSP.

For the given supergraph $H = (V, C)$, assume $|V| = n$, $C = \{T_1, T_2, \ldots, T_m\}$ and $V = \cup_{i=1}^{m}\{T_i\}$. Denote $k = \max\{|T_i| \mid T_i \in C\}$. Assume there is a submodular function $f(\cdot) : 2^V \to R_+$ with $f(\emptyset) = 0$, where $f(X)$ means the covering cost of X for any subset X of V.

Firstly we define a binary variable $\zeta(X)$ for any subset $X \subseteq V$ as follows:

$$\zeta(X) = \begin{cases} 1, & \text{if } X \text{ is selected to cover some hyperedges,} \\ 0, & \text{otherwise.} \end{cases}$$

Therefore, the integer linear program for the SHSP is

$$\begin{aligned} \min & \sum_{X\subseteq V} f(X)\zeta(X) & & \\ \text{s.t.} & \sum_{\substack{X\subseteq V\\ X\cap T_i\neq\emptyset}} \zeta(X) \geq 1, & & \forall T_i \in C, \\ & \zeta(X) \in \{0, 1\}, & & \forall X \subseteq V. \end{aligned} \tag{3}$$

In the above program, the first constraint ensures that each hyperedge $T_i \in C$ is covered by some vertex subset X. Since $f(\cdot)$ is a submodular function, there must exist a unique subset X of V such that $\zeta(X) = 1$ in the optimal solution of (3).

Recall that $\hat{f}(\boldsymbol{\chi}_X) = f(X)$, the convex programming for the SHSP is:

$$\begin{aligned} &\min \hat{f}(\boldsymbol{x}) \\ &\text{s.t.} \sum_{u\in T_i} \boldsymbol{x}(u) \geq 1, \qquad \forall T_i \in C, \\ &\qquad \boldsymbol{x}(u) \in \{0,1\}, \qquad \forall u \in V. \end{aligned} \tag{4}$$

The relaxation of (4) is:

$$\begin{aligned} &\min \hat{f}(\boldsymbol{x}) \\ &\text{s.t.} \sum_{u\in T_i} \boldsymbol{x}(u) \geq 1, \qquad \forall T_i \in C, \\ &\qquad \boldsymbol{x}(u) \geq 0, \qquad \forall u \in V. \end{aligned} \tag{5}$$

3.1. A Rounding Algorithm for the SHSP

For a submodular function $f(\cdot)$, we define a set function $f^\circ(\cdot) : 2^V \to R_+$ as

$$f^\circ(X) = \min\{f(Z) \mid X \subseteq Z \subseteq V\}, \qquad \forall X \subseteq V.$$

Then $f^\circ(\cdot)$ is monotone and submodular [10]. Moreover, this function $\hat{f}^\circ(\cdot)$ has the following properties.

Lemma 3. ([6])

(1) suppose X° is the exactly minimal subset Z satisfying $X \subseteq Z \subseteq V$ and $f(Z) = f^\circ(X)$. Then $f^\circ(X) = f(X^\circ)$ and $f(X) \geq f^\circ(X)$ for every subset $X \subseteq V$;
(2) for each $\boldsymbol{x} \in R^n_+$, $\hat{f}(\boldsymbol{x}) \geq \hat{f}^\circ(\boldsymbol{x})$;
(3) $\hat{f}^\circ(\cdot)$ is monotonically increasing and positively homogeneous.

Since the program (5) can be solved in polynomial time via the ellipsoid method, we denote $\boldsymbol{x}^* \in R^n_+$ an optimal solution of (5). Based on the program (5), we give the following rounding algorithm.

Algorithm 1 (H). **Step 0:** *Solve the program* (5) *and denote* $\boldsymbol{x}^* \in R^n_+$ *as the optimal solution.*
Step 1: *Define the subset* $A \subseteq V$ *by*

$$A = \{u \in V \mid \boldsymbol{x}^*(u) \geq \frac{1}{k}\}.$$

Step 2: *Define* A° *be the exactly minimal subset of* V *such that* $A \subseteq A^\circ$ *and* $f(A^\circ) = f^\circ(A)$.
Step 3: *Return* A°.

Recall that $k = \max\{|T_i| \mid T_i \in C\}$. For every hyperedge $T_i \in C$, since $\sum_{u\in T_i} \boldsymbol{x}(u) \geq 1$, at least one of $\boldsymbol{x}^*(u)$ must be greater than $\frac{1}{k}$, which ensures A is a hitting set. Since $A \subseteq A^\circ$, A° is also a hitting set. Moreover, the hitting set A° can be obtained in polynomial time by submodular function minimization theory. Therefore, Algorithm 1 can be implemented in polynomial time and the obtained A° is a hitting set. Finally, we analyze the approximation ratio of Algorithm 1.

Theorem 1. *With reference to* $f(\cdot)$, $f^\circ(\cdot)$ *and* A° *defined above, then we have* $f(A^\circ) \leq kf(M)$ *for any hitting set* M*, where* $k = \max\{|T_i| \mid T_i \in C\}$.

Proof. Since $\boldsymbol{x}^* \in R^n_+$ is an optimal solution of (5) and $\hat{f}(\boldsymbol{\chi}_M) = f(M)$ for any hitting set M, we find

$$\hat{f}(\boldsymbol{x}^*) \leq \hat{f}(\boldsymbol{\chi}_M) = f(M).$$

Lemma 3 (2) guarantees

$$\hat{f}^\circ(\boldsymbol{x}^*) \leq \hat{f}(\boldsymbol{x}^*)$$

and hence

$$\hat{f}^{\circ}(\boldsymbol{x}^{*}) \leq f(M). \tag{6}$$

The fact $\boldsymbol{\chi}_A \leq k\boldsymbol{x}^*$ and Lemma 3 (3) ensure that

$$\hat{f}^{\circ}(\boldsymbol{\chi}_A) \leq \hat{f}^{\circ}(k\boldsymbol{x}^{*}) = k\hat{f}^{\circ}(\boldsymbol{x}^{*}).$$

Note that $\hat{f}^{\circ}(\boldsymbol{\chi}_A) = f^{\circ}(A) = f(A^{\circ})$. Then we have

$$f(A^{\circ}) \leq k\hat{f}^{\circ}(\boldsymbol{x}^{*}). \tag{7}$$

Combining (6) with (7), we obtain $f(A^{\circ}) \leq kf(M)$ as desired. □

3.2. A Primal-dual Algorithm

In this subsection, we give a primal-dual algorithm for the SHSP. For any subset $X \subseteq V$, we denote $\mathcal{S}_X = \{T_i \in C \mid T_i \cap X \neq \emptyset\}$ and we call X is a hitting set if $\mathcal{S}_X = C$.

Recall that for any given vector $\boldsymbol{p} \in R^n_+$, $\hat{f}(\boldsymbol{p})$ is equal to the optimal value of (2). Combining (2) with (5), we consider the following relaxation program for the SHSP :

$$\begin{aligned} \min & \sum_{X\subseteq V} f(X)\cdot\zeta(X) & & \\ \text{s.t.} & \sum_{u\in T_i} \boldsymbol{x}(u) \geq 1, & & \forall T_i \in C, \\ & \sum_{\substack{X\subseteq V\\ u\in X}} \zeta(X) = \boldsymbol{x}(u), & & \forall u \in V, \\ & \zeta(X) \geq 0, & & \forall X \subseteq V. \end{aligned} \tag{8}$$

We do not need to add the constraint that $\boldsymbol{x}(u) \geq 0$ for the second constraint of (8) ensures it for all $u \in V$. And the dual program for (8) is:

$$\begin{aligned} \max & \sum_{T_i\in C} y(T_i) & & \\ \text{s.t.} & \sum_{\substack{T_i\in C\\ u\in T_i}} y(T_i) = \boldsymbol{z}(u), & & \forall u \in V, \\ & \boldsymbol{z} \in P(f), & & \\ & y(T_i) \geq 0, & & \forall T_i \in C. \end{aligned} \tag{9}$$

Before proposing the primal-dual algorithm for the SHSP, we give some necessary facts.

Given a subbase $\boldsymbol{z} \in P(f)$. Let $(f-\boldsymbol{z})(\cdot) : 2^V \to R_+$ be defined by

$$(f-\boldsymbol{z})(X) = f(X) - \boldsymbol{z}(X), \qquad \forall X \subseteq V.$$

Then $(f-\boldsymbol{z})(\cdot)$ is a submodular function. Observe that the definition of $(f-\boldsymbol{z})(\cdot)$ is related to the modularity of $\boldsymbol{z}(\cdot)$. So we get the function $(f-\boldsymbol{z})(\cdot)$ for any subbase $\boldsymbol{z} \in P(f)$ and there is a proposition for the function.

Proposition 2. *Given a subbase $\boldsymbol{z} \in P(f)$, there exists the unique maximal subset X of V for the function $(f-\boldsymbol{z})(\cdot)$ such that $f(X) = \boldsymbol{z}(X)$.*

Proof. Note that $\boldsymbol{z} \in P(f)$, it is clear that

$$(f-\boldsymbol{z})(X) \geq 0, \qquad \forall X \subseteq V.$$

Moreover, we get that $\min_{X\subseteq V}(f-z)(X) = (f-z)(\emptyset) = 0$. This implies that the existence of X. According to Proposition 1, we see that for every pair of minimizers X, Y of function $(f-z)(\cdot)$, $X\cup Y$ is also a minimizer. Thus, there exists the unique maximal subset $X \subseteq V$ for the function $(f-z)(\cdot)$ such that $f(X) = z(X)$. □

Lemma 4. *Let $z \in P(f)$ and A be the unique maximal subset of V satisfying $f(A) = z(A)$. Assume $C\backslash\mathcal{S}_A \neq \emptyset$ and a hyperedge $T_j \in C\backslash\mathcal{S}_A$. If we define*

$$\alpha = \min_{\substack{X\subseteq V\\ \chi_{T_j}(X)\neq 0}} \frac{f(X)-z(X)}{\chi_{T_j}(X)}$$

and

$$z' = z + \alpha\cdot\chi_{T_j},$$

then we have

(1) $z' \in P(f)$;

(2) furthermore, we define A' as the maximal subset of V such that $f(A') = z'(A')$. Then we have $A \subsetneq A'$.

Proof. To prove $z' \in P(f)$, it suffices to prove $z'(X) \le f(X)$ for every subset X of V. The assumption $z \in P(f)$ implies that

$$z(X) \le f(X), \qquad \forall X \subseteq V.$$

On the one hand, for every subset X of V such that $\chi_{T_j}(X) = 0$, we have

$$z'(X) = z(X) + \alpha\cdot 0 = z(X) \le f(X).$$

On the other hand, for every subset X of V such that $\chi_{T_j}(X) \neq 0$, we have

$$z'(X) = z(X) + \alpha\cdot\chi_{T_j}(X) \le z(X) + \frac{f(X)-z(X)}{\chi_{T_j}(X)}\cdot\chi_{T_j}(X) = f(X),$$

where the inequality follows from the definition of α. The result follows.

Next we prove $A \subsetneq A'$. The assumption $T_j \in C\backslash\mathcal{S}_A$ indicates $A\cap T_j = \emptyset$. The definition of z' implies $z'(u) = z(u)$ for every element $u \in A$. Moreover, we get $z'(A) = z(A) = f(A)$. The definition of A' implies $A \subseteq A'$. Suppose Z is a subset of V which satisfies both $\chi_{T_j}(Z) \neq 0$ and

$$\alpha = \frac{f(Z)-z(Z)}{\chi_{T_j}(Z)}.$$

Then we have $f(Z) = z'(Z)$. The definition of A' implies $Z \subseteq A'$. Since $\chi_{T_j}(Z) \neq 0$, we have $T_j\cap Z \neq 0$. Combining $A\cap T_j = \emptyset$ with $T_j \cap Z \neq 0$, we obtain $A \nsubseteq Z$. Since $A \nsubseteq Z, A \subseteq A'$ and $Z \subseteq A'$, it is immediate that $A \subsetneq A'$. □

Based on the dual program (9), we give the following primal-dual algorithm.

Algorithm 2 (H). **Step 0:** *Set $t = 1$. Initialize $y_1 = 0, z_1 = 0$. Define A_1 as the unique maximal subset of V such that $f(A_1) = z_1(A_1)$.*

Step 1: *If $C\backslash\mathcal{S}_{A_t} = \emptyset$, go to step 2. If not, select a hyperedge $T_j \in C\backslash\mathcal{S}_{A_t}$.*

Define the real number α_t by

$$\alpha_t = \min_{\substack{X\subseteq V\\ \chi_{T_j}(X)\neq 0}} \frac{f(X)-z_t(X)}{\chi_{T_j}(X)}$$

Define y_{t+1} by

$$y_{t+1}(T_i) = \begin{cases} y_t(T_i) + \alpha_t, & \textit{if } T_i = T_j\,, \\ y_t(T_i), & \textit{otherwise} \end{cases}$$

and let

$$z_{t+1} = z_t + \alpha_t \cdot \boldsymbol{\chi}_{T_j}.$$

Define A_{t+1} to be the unique maximal subset of V such that $z_{t+1}(A_{t+1}) = f(A_{t+1})$.
Set $t = t + 1$.

Step 2: *Let $T = t$ and $A = A_T$. Return A.*

To obtain the ratio of Algorithm 2, we first prove the following lemmas.

Lemma 5. *Algorithm 2 can be implemented in polynomial time.*

Proof. Combining the definition of A with Lemma 4 (2), we get that the algorithm iterates after at most $|V| + 1$ times. That is, T is at most $|V| + 1$. At every iteration, we can compute α_t in polynomial time [11]. The Proposition 2 guarantees there exists the unique maximal $A_{t+1} \subseteq V$ such that $z_{t+1}(A_{t+1}) = f(A_{t+1})$ and we can compute it in polynomial time [7]. Therefore, Algorithm 2 can be implemented in polynomial time. □

Algorithm outputs $\{(y_1, z_1), (y_2, z_2), \cdots, (y_T, z_T)\}$. Next we will prove it is a feasible region of the dual program (9).

Lemma 6. *For each $t \in \{1, 2, \cdots, T\}$, (y_t, z_t) is a feasible solution of the dual program* (9).

Proof. Need to show for each $t \in \{1, 2, \cdots, T\}$, (y_t, z_t) satisfies all constrains of (9). We prove this result by using induction on t.

Since $(y_1, z_1) = \mathbf{0}$ and $f(\cdot)$ is a nonnegative function, we obtain (y_1, z_1) is a feasible solution of (9). Assume that this lemma holds when $t = k$. Suppose $t = k + 1$ and $T_j \in C \backslash \mathcal{S}_{A_k}$. Firstly, according to Lemma 4 (1), it is obvious that $z_{k+1} \in P(f)$. Combining $z_k \in P(f)$ with the definition of α_k, we have $\alpha_k \geq 0$, which implies $y_{k+1} \geq 0$. Next, we prove that for each $u \in V$, (y_{k+1}, z_{k+1}) satisfies

$$\sum_{T_i \in C : u \in T_i} y_{k+1}(T_i) = z_{k+1}(u). \tag{10}$$

Since (y_k, z_k) is a feasible solution of (9), we have

$$\sum_{T_i \in C : u \in T_i} y_k(T_i) = z_k(u), \qquad \forall u \in V. \tag{11}$$

Note that $V = T_j \cup (V \backslash T_j)$ and $z_{k+1} = z_k + \alpha_k \cdot \boldsymbol{\chi}_{T_j}$. For each $u \in V \backslash T_j$, we obtain

$$z_{k+1}(u) = z_k(u) \tag{12}$$

and

$$\sum_{T_i \in C : u \in T_i} y_{k+1}(T_i) = \sum_{T_i \in C : u \in T_i} y_k(T_i). \tag{13}$$

From (11)-(13), it is immediate that (10) holds for each $u \in V \backslash T_j$. For each $u \in T_j$, we have

$$z_{k+1}(u) - z_k(u) = \alpha_k$$

and

$$\sum_{T_i \in C : u \in T_i} y_{k+1}(T_i) - \sum_{T_i \in C : u \in T_i} y_k(T_i) = y_{k+1}(T_j) - y_k(T_j) = \alpha_k,$$

which implies that (10) holds for each $u \in T_j$. Therefore, we have (10) for each $u \in V$. □

We now analyze the approximation ratio of Algorithm 2.

Theorem 2. *Algorithm 2 is a k-approximation algorithm, where* $k = \max\{|T_i| \mid T_i \in C\}$.

Proof. Algorithm 2 returns a hitting set A and the final feasible solution (y_T, z_T) for the dual program (9). Let OPT denote the optimal objective value for the SHSP. In what follows, we prove that

$$f(A) \le k \cdot OPT.$$

Based on Lemma 6 and the dual program (9), we get

$$f(A) = z_T(A) = \sum_{u \in A} z_T(u) = \sum_{u \in A} \sum_{\substack{T_i \in C \\ u \in T_i}} y_T(T_i). \tag{14}$$

Using the definitions of k and A, it is immediate that

$$\sum_{u \in A} \sum_{\substack{T_i \in C \\ u \in T_i}} y_T(T_i) \le k \sum_{T_i \in C} y_T(T_i). \tag{15}$$

The final feasible solution (y_T, z_T) for (9) indicates that

$$\sum_{T_i \in C} y_T(T_i) \le OPT. \tag{16}$$

From (14)-(16), we get $f(A) \le k \cdot OPT$. □

Acknowledgment

The authors would like to thank Professor Ding-Zhu Du for his many valuable advices during their study of approximation algorithm.

This work is supported by the NSF of China (No. 11971146), the NSF of Hebei Province (No. A2019205089, No. A2019205092), Hebei Province Foundation for Returnees (CL201714) and Overseas Expertise Introduction Program of Hebei Auspices (25305008).

References

1. D. Du, K. Ko, X. Hu, *Design and analysis of approximation Algorithms* (Springer, New York, 2011).
2. P. Wan, D. Du, P. Pardalos, W. Wu, Greedy approximations for minimum submodular cover with submodular cost. J. *Comput Optim Appl* 45(2)(2010), 463–474.
3. C. Koufogiannakis, N. E. Young, Greedy Δ–approximation algorithm for covering with arbitrary constrains and submodular costs. J. *Algorithmica* 66(1)(2013), 133–141.
4. D. S. Hochbaum, Submodular problems–approximations and algorithms. *Technical Report arXiv*(2010), 1010.1945.
5. N. Kamiyama, A note on submodular function minimization with covering type linear constraints. J. *Algorithmica* 80(2018), 2957–2971.
6. S. Iwata, K. Nagano, Submodular function minimization under covering constraints. In: Proceedings of the 50th annual symposium on foundations of computer science, 2009, pp 671–680.
7. L. Lovász, Submodular functions and convexity. In: A. Bachm, M. Grtschel, B. Korte (Eds.), Mathematical Programing the State of the Art. Springer, Berlin, 1983, pp 235–237.

8. J. Edmonds, Submodular functions, matroids, and certain polyhedra. In: R. Guy, H. Hanam, N. Sauer, J. Schonheim(Eds.), Combinatorial Structures and Their Applications, Proc. 1969 Calgary Conference, Gordon and Breach, 1970, pp 69–87.

9. Y. Li, D. Du, N. Xiu, D. Xu, Improved approximation algorithms for the facility location problems with linear/submodular penalties. J. *Algorithmica* 73(2)(2015), 460–482.

10. S. Fujishige, *Submodular functions and optimization* (Elsevier, Amsterdam, 2005).

11. K. Murota, Discrete Convex Analysis, volume 10 of SIAM monographs on discrete mathematics and applications. *SIAM*(2003).

Proceedings of the 8th International Conference on Theory and Applications of Models of Computation, volume 6648 of Lecture Notes in Computer Science, 2011, pp 133–141.