Math. Program., Ser. A 102: 167–183 (2005)

Digital Object Identifier (DOI) 10.1007/s10107-004-0510-2

Liang Zhao · Hiroshi Nagamochi · Toshihide Ibaraki

Greedy splitting algorithms for approximating multiway partition problems

Received: February 7, 2002 / Accepted: January 30, 2004
Published online: 28 April 2004 – © Springer-Verlag 2004

Abstract. Given a system (V, T, f, k), where V is a finite set, $T \subseteq V$, $f : 2^V \to \mathbf{R}$ is a submodular function and $k \geq 2$ is an integer, the general *multiway partition problem* (MPP) asks to find a k-partition $\mathcal{P} = \{V_1, V_2, \ldots, V_k\}$ of V that satisfies $V_i \cap T \neq \emptyset$ for all i and minimizes $f(V_1) + f(V_2) + \cdots + f(V_k)$, where $\mathcal{P}$ is a k-partition of V if (i) $V_i \neq \emptyset$, (ii) $V_i \cap V_j = \emptyset$, $i \neq j$, and (iii) $V_1 \cup V_2 \cup \cdots \cup V_k = V$ hold. MPP formulation captures a generalization in submodular systems of many NP-hard problems such as *k-way cut*, *multiterminal cut*, *target split* and their generalizations in hypergraphs. This paper presents a simple and unified framework for developing and analyzing approximation algorithms for various MPPs.

Key words. Approximation algorithm – Hypergraph partition – k-way cut – Multiterminal cut – Multiway partition problem – Submodular function

1. Introduction

Let V be a finite set. A function $f : 2^V \to \mathbf{R}$ is said to be *submodular* if $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$ holds for all $X, Y \subseteq V$. It is *nonnegative* (resp., *symmetric*) if $f(X) \geq 0$ (resp., $f(X) = f(V - X)$) holds for all $X \subseteq V$, and *monotone* if $f(X) \leq f(Y)$ holds for all $X \subseteq Y \subseteq V$. We call (V, f) a *submodular system* if function f is submodular. Analogous notations are also used.

The *multiway partition problem* (MPP) is defined as follows.

Problem 1 (MPP). Given a system (V, T, f, k), where (V, f) is a submodular system, $T \subseteq V$ called the target set, and k is an integer with $2 \leq k \leq |T|$,

$$\begin{aligned} \text{minimize} \quad & f(V_1) + f(V_2) + \cdots + f(V_k) \\ \text{subject to} \quad & V_1 \cup V_2 \cup \cdots \cup V_k = V, && (1) \\ & V_i \cap V_j = \emptyset, \quad 1 \leq i < j \leq k, && (2) \\ & V_i \cap T \neq \emptyset, \quad i = 1, 2, \ldots, k. && (3) \end{aligned}$$

L. Zhao: Dept. Information Science, Faculty of Engineering, Utsunomiya University, Yoto 7-1-2, Utsunomiya, 321-8585, Japan, e-mail: zhao@is.utsunomiya-u.ac.jp

H. Nagamochi: Dept. Information and Computer Sciences, Toyohashi University of Technology, Toyohashi, Aichi, 441-8580, Japan, e-mail: naga@ics.tut.ac.jp

T. Ibaraki: Dept. Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan, e-mail: ibaraki@i.kyoto-u.ac.jp

Mathematics Subject Classification (1991): 20E28, 20G40, 20C20

A family $\mathcal{P} = \{V_1, V_2, \ldots, V_k\}$ of nonempty subsets of V satisfying (1) and (2) is called a *k-partition* (of V). The *weight* of $\mathcal{P}$ is defined by $f(\mathcal{P}) \triangleq \sum_{i=1}^{k} f(V_i)$. A feasible solution to MPP, i.e., a k-partition satisfying (3), is called a *k-target-split* (of V, with respect to T). We assume that function f is given by an oracle which returns the value $f(S)$ in at most θ time for any $S \subseteq V$.

1.1. MPP with no special target (MPP-NT)

We first consider the problem of finding a minimum k-partition, i.e., a special case of MPP with $T = V$. We denote it by MPP-NT (*MPP with No special Target*). The next problem classes are treated.

Problem 2 (k-PPSS). Given a nonnegative submodular system (V, f), the k-partition problem in submodular system (k-PPSS) is to find a minimum weight k-partition.

Problem 3 (k-PPSSS). The k-partition problem in symmetric submodular system (k-PPSSS) is k-PPSS with symmetric f.

Problem 4 (k-PPMSS). The k-partition problem in monotone submodular system (k-PPMSS) is k-PPSS with monotone f.

We further treat three types of partition problems in hypergraphs. A *hypergraph H* is a pair (V, E) of a set V of *vertices* and a set E of *hyperedges*, where hyperedges are nonempty subsets of V. The *degree* of a hyperedge is defined as its cardinality. A hyperedge $e = \{v_1, \ldots, v_j\}$ may also be treated as the set $\{v_1, \ldots, v_j\}$ of vertices, where each v_i is called an *endpoint* of e. In general, we allow the existence of *multiple hyperedges*, i.e., hyperedges with the same endpoints. Thus E should be viewed as a multiset that allows the existence of multiple members. Graphs are hypergraphs in which each hyperedge (i.e., edge) has degree 2. Given a hypergraph $H = (V, E)$ with a weight function $w : E \to \mathbf{R}^+$ on hyperedges (where $\mathbf{R}^+$ denotes the set of nonnegative numbers), we define two functions $w_{\text{ex}}, w_{\text{in}} : 2^V \to \mathbf{R}^+$ by

$$w_{\text{ex}}(S) = \sum_{\emptyset \subset e \cap S \subset e} w(e), \quad w_{\text{in}}(S) = \sum_{e \subseteq S} w(e), \quad S \subseteq V. \tag{4}$$

It is well-known and can be easily verified that function w_{ex} (called the *cut function*) is symmetric and submodular. It is also easy to see that function $w_{\text{ex}} + w_{\text{in}}$ is monotone and submodular, whereas function $-w_{\text{in}}$ is submodular. Denote the weight of a subset E' of hyperedges by $w(E') \triangleq \sum_{e \in E'} w(e)$. We define the next problems.

Definition 1. *Given a hypergraph $H = (V, E)$ with a weight function $w : E \to \mathbf{R}^+$ on hyperedges, the k-partition problem in hypergraphs of type i* (k-PPH-Ti) *is MPP with system (V, V, f_i, k), $i = 1, 2, 3$, where functions f_i are defined as*

$$f_1 \triangleq \frac{w(E)}{k} - w_{\text{in}}, \quad f_2 \triangleq w_{\text{ex}} \quad \textit{and} \quad f_3 \triangleq w_{\text{ex}} + w_{\text{in}} - \frac{w(E)}{k}.$$

Remark. Let $\mathcal{P} = \{V_1, V_2, \ldots, V_k\}$ be a k-partition of V, and let e be a hyperedge. Let

$$p_e = \left|\{i \mid e \cap V_i \neq \emptyset\}\right|.$$

Clearly $p_e \geq 1$. It is easy to see that, if $p_e = 1$ then the weight of e is not counted in any $f_i(\mathcal{P})$, $i = 1, 2, 3$; otherwise it is counted *once*, p_e *times* and $p_e - 1$ *times* in $f_1(\mathcal{P})$, $f_2(\mathcal{P})$ and $f_3(\mathcal{P})$, respectively. Thus three problems k-PPH-Ti, $i = 1, 2, 3$, are equivalent if H is restricted to graphs, which we call the *k-partition problem in graphs* (k-PPG). In general hypergraphs, if k is fixed to 2, they are also equivalent (called the *minimum cut problem* in hypergraphs). □

Obviously the next inclusion relationship holds for the above problem classes.

$$k\text{-PPG} \subset k\text{-PPH-T2} \subset k\text{-PPSSS} \subset k\text{-PPSS}.$$

Notice that k-PPH-T3 is equivalent to a k-PPMSS with system $(V, V, w_{\text{ex}}+w_{\text{in}}, k)$ (in the meaning of exact solution). Thus we may also say

$$k\text{-PPG} \subset k\text{-PPH-T3} \subset k\text{-PPMSS} \subset k\text{-PPSS}.$$

Problem k-PPH-T2 was introduced by Lawler [17], who suggested many applications such as network analysis (where k-PPG was considered), information storage and retrieval and others. Later Pulleyblank [24] (see [2]) formulated the *k-way cut problem* (also called the *k-cut problem*) in connection with generating cutting planes for the traveling salesman problem (TSP). Given a graph with nonnegative weights on edges, the k-way cut problem is to find a minimum weight subset of edges whose removal leaves the graph with k connected components. Obviously, this is equivalent to k-PPG (hence we will not distinguish them). For other applications, see e.g., [18, 19, 30].

Goldschmidt and Hochbaum [8] showed that k-PPG is NP-hard even for unit edge weights. This implies the NP-hardness of all the above problem classes. They also showed that k-PPG is solvable in $O(n^{k^2})$ time. (Throughout this paper, we use n and m to denote $|V|$ and the number of hyperedges/edges respectively.) Queyranne [26] claimed that k-PPSSS is solvable in $O(n^{k^2}\theta)$ time.

Problem 2-PPSS can be solved in $O(n^3\theta)$ time ([25]). This implies that the minimum cut problem in hypergraphs can be solved in $O(n^3 D)$ time, where D is the sum of degrees of hyperedges. This is improved to $O(n^2 \log n + nD)$ by Klimmek and Wagner [16]. A minimum cut in graphs can be found in $O(mn + n^2 \log n)$ time ([22]).

Problems k-PPH-T1 and k-PPH-T3 arise from network reliability analysis ([30]) and VLSI design ([4]), respectively. On the other hand, we note that k-PPMSS can be viewed as a partition problem in polymatroid, where f is the rank function. For more about polymatroid, see e.g., [15].

Due to the NP-hardness, it is of interest to develop *approximation algorithms*. An algorithm (for MPP with nonnegative optimum) is said a *ρ-approximation algorithm* if it always delivers a feasible solution whose weight is at most ρ times of the optimum. Value ρ is called the *performance guarantee*, or simply *guarantee*.

Saran and Vazirani [27] proposed two approximation algorithms for k-PPG (see also [29]). One is based on the *cut tree* structure of graphs (also known as the *Gomory-Hu tree*, see [7]). The other *greedily* increases the size of partitions by one in each iteration.

Table 1. Approximation results for MPP-NTs.

problem class	guarantee	running time	reference
k-PPG	$2 - \frac{2}{k}$	$O(kn(m + n \log n))$	[13, 27], this paper
k-PPSSS	$2 - \frac{2}{k}$	$O(kn^3\theta)$	[26], this paper
k-PPSS	$k - 1$	$O(kn^3\theta)$	this paper
k-PPMSS	$2 - \frac{2}{k}$	$O(kn^3\theta)$	this paper
k-PPH-T1	$\min\{k, d^+_{\max}\}(1 - \frac{1}{k})$ †	$O(kn(n \log n + D))$	this paper
k-PPH-T{2,3}	$2 - \frac{2}{k}$	$O(kn(n \log n + D))$	this paper

† $d^+_{\max}$: the maximum degree of hyperedges with *positive* weight.

By a sophisticated proof using the cut tree structure, they showed that both algorithms have a guarantee of $2 - \frac{2}{k}$. Later Kapoor [13] pointed out that the greedy algorithm in [27] can be implemented in $O(kn(m + n \log n))$ time (on the other hand, the cut-tree based algorithm requires $O(n^2(m + n \log n))$ time).

Queyranne [26] claimed that the greedy algorithm in [13, 27] can be extended to approximate k-PPSSS, enjoying the same performance guarantee of $2 - \frac{2}{k}$. Narayanan, Roy and Patkar [23] showed (without showing the running time) that, k-PPH-T1 and k-PPH-T3 can be approximated within factors $d_{\max}(1 - \frac{1}{n})$ and $2 - \frac{2}{n}$ respectively, where $d_{\max}$ is the maximum degree of hyperedges.

We note that, all the proofs in [23, 26, 27] use specific structures (cut tree or the so-called *principal partition*), which are rather complicated and cannot be applied to general submodular systems (see [1]). In this paper, we consider the greedy algorithm. We first show a simple lemma (Lemma 1) that holds for all submodular systems, then derive guarantees for various MPPs by easy calculations. In particular, the same or improved results compared to [13, 23, 27, 26] can be obtained in a much simpler way. Table 1 shows the main approximation results (due to the greedy algorithm).

1.2. General MPPs

Similarly as in Subsection 1.1, we treat the next classes of the MPP. Given a nonnegative submodular system (V, f) with a target set $T \subseteq V$, the *k-target-split problem in submodular systems* (k-TPSS) is to find a minimum weight k-target-split. Problem k-TPSSS (resp., k-TPMSS) is k-TPSS with symmetric (resp., monotone) f. Problem k-TPH-T2 is a special case of k-TPSSS, in which V and f are respectively the vertex set and the cut function of the input hypergraph. Problems k-TPH-T1 and k-TPH-T3 are MPPs with systems $(V, T, \frac{w(E)}{k} - w_{\text{in}}, k)$ and $(V, T, w_{\text{ex}} + w_{\text{in}} - \frac{w(E)}{k}, k)$, respectively. Problems k-TPH-Ti ($i = 1, 2, 3$) reduce to k-TPG when restricted to graphs. The inclusion relationships among these problem classes are

$$k\text{-TPG} \subset k\text{-TPH-T2} \subset k\text{-TPSSS} \subset k\text{-TPSS},$$
$$k\text{-TPG} \subset k\text{-TPH-T3} \subset k\text{-TPMSS} \subset k\text{-TPSS}.$$

Problem k-TPG with $|T| = k$ is also known as the *multiterminal cut problem* (k-MCP). Dahlhaus, Johnson, Papadimitriou, Seymour and Yannakakis [6] showed that, k-MCP with any fixed $k \geq 3$ is NP-hard (actually MAXSNP-hard) even for unit edge weights. They also provided a simple $(2 - \frac{2}{k})$-approximation algorithm. Later Călinescu, Karloff and Rabani [3] gave a novel geometric relaxation for k-MCP to obtain a $(\frac{3}{2} - \frac{1}{k})$-approximation algorithm. Karger, Klein, Stein, Thorup and Young [14] further showed that the performance guarantee can be improved to 1.3438.

Garg, Vazirani and Yannakakis [11] considered a *vertex weighted* version of k-MCP, where instead of edges, non-target vertices have nonnegative weights, and the objective is to find a minimum weight subset of non-target vertices whose removal separates each pair of targets. It is easy to reduce this problem to k-TPH-T1 with $|T| = k$, by replacing each non-target vertex with a hyperedge of the same weight. Based on an LP relaxation, Garg et al gave a $(2 - \frac{2}{k})$-approximation algorithm. On the other hand, Chopra and Owen [4] gave several formulations for k-TPH-T3 with $|T| = k$, and provided some computational experiment results.

While all the previous papers treat k-MCP or its generalization in hypergraphs, Maeda, Nagamochi and Ibaraki [21] considered the first target-split problem k-TPG. They showed that it can be approximated within a factor of $2 - \frac{2}{k}$. As far as we know, there are no more results available for any of the other versions of target-split problems.

We note that the relaxations in [3, 11] exploit the fact $|T| = k$, i.e., there are exactly k targets. They do not yield polynomial time algorithm for the target split problems. In this paper, we give a greedy approach for approximating MPP by modifying the one for MPP-NT. We show a simple key lemma (Lemma 4), from which an immediate result is that the same guarantees as for MPP-NTs hold for the corresponding MPPs.

2. Greedy splitting algorithm (GSA) for MPP-NT

2.1. Greedy splitting approach

Let us first give a general description of the greedy splitting approach. The idea is simple. We start at the 1-partition $\mathcal{P}_1 = \{V\}$. In the i^{th} iteration, $i \geq 1$, we construct an $(i + 1)$-partition $\mathcal{P}_{i+1}$ by splitting some member of the previously obtained i-partition $\mathcal{P}_i$. We halt when $i = k - 1$ holds. Since it is desired to get a solution of small weight, we want to minimize the weight increase $f(\mathcal{P}_{i+1}) - f(\mathcal{P}_i)$ (called the *splitting weight*) for each i. We will show that this can be done in polynomial time.

2.2. Algorithm description and main lemma

We first show that a variant of 2-PPSS can be solved in polynomial time.

Theorem 1 (Queyranne [25]). *Given a symmetric submodular system (V, g), where $n = |V| \geq 2$ holds, a nonempty proper subset S^* of V such that $g(S^*)$ is minimum can be found in $O(n^3\theta)$ time, where θ is the time bound of the oracle for g.* □

Theorem 2. *Given a submodular system (V, f) and a set $W \subseteq V$ with $|W| \geq 2$, a nonempty proper subset S^* of W such that $f(S^*) + f(W - S^*)$ is minimum can be found in $O(|W|^3\theta)$ time, where θ is the time bound of the oracle for f.*

Input: A submodular system (V, f) and an integer $k \geq 1$.
Output: A k-partition $\mathcal{P}_k$ of V.

```
1  P_1 ← {V}
2  for i = 1, ..., k − 1 do
3      (S_i, W_i) ← argmin {f(S) + f(W−S) − f(W) | ∅ ⊂ S ⊂ W, W ∈ P_i}
4      P_{i+1} ← (P_i − {W_i}) ∪ {S_i, W_i − S_i}
5  end   /* for */
```

Fig. 1. Greedy splitting algorithm (GSA) for MPP-NT.

Proof. Define a function $g: 2^W \to \mathbf{R}$ by $g(S) = f(S) + f(W-S)$, $S \subseteq W$. Notice that (W, g) is a symmetric submodular system. Thus the theorem holds by Theorem 1. □

We now present the *greedy splitting algorithm* (GSA) for MPP-NT, see Fig. 1.

We start at $\mathcal{P}_1 = \{V\}$. In the i^{th} iteration, we compute a pair (S_i, W_i) that minimizes the splitting weight $f(S) + f(W-S) - f(W)$ over all (S, W) satisfying $\emptyset \subset S \subset W$ and $W \in \mathcal{P}_i$. $\mathcal{P}_{i+1}$ is obtained by replacing W_i with S_i and $W_i - S_i$. Obviously, for each $i = 1, 2, \ldots, k$, the weight of $\mathcal{P}_i$ is

$$f(\mathcal{P}_i) \;=\; f(V) + \sum_{j=1}^{i-1}\Big(f(S_j) + f(W_j - S_j) - f(W_j)\Big).$$

Clearly, the output $\mathcal{P}_k$ is a k-partition of V. Let us consider the running time of GSA. For any (fixed) $W \subseteq V$, Theorem 2 shows that we can minimize $f(S) + f(W-S)$, hence $f(S) + f(W-S) - f(W)$, in $O(|W|^3\theta)$ time over $\emptyset \subset S \subset W$. Hence Line 3 can be done in $\sum_{W\in\mathcal{P}_i} O(|W|^3\theta) = O(\sum_{W\in\mathcal{P}_i} |W|^3\theta) = O(n^3\theta)$ time. This implies that the running time of GSA is $O(kn^3\theta)$.

Theorem 3. *Given a submodular system (V, f) and an integer $k \geq 1$, GSA finds an i-partition $\mathcal{P}_i$ of V for all $i = 1, 2, \ldots, k$ in a total time of $O(kn^3\theta)$.* □

Remark. We may implement GSA by computing a nonempty $S \subset W$ minimizing $f(S)+f(W-S)$ for each $W \in \mathcal{P}_{i+1} - \mathcal{P}_i$ at the end of the i^{th} iteration to avoid duplicate computations. We may also use an efficient 2-partition algorithm instead of calling the oracle. E.g., Line 3 can be done in $O(mn + n^2 \log n)$ time for k-PPG by employing the mincut algorithm [22]. This was first pointed out by Kapoor [13]. □

To derive the performance guarantees, we first show a technical lemma.

Lemma 1 (Main lemma). *Let $\mathcal{P}_i$ be the i-partition of V found by GSA in the $(i-1)^{th}$ iteration, $1 \leq i \leq k$. For any i-partition $\mathcal{P} = \{V_1, V_2, \ldots, V_i\}$ of V, it holds that*

$$f(\mathcal{P}_i) \;\leq\; \sum_{j=1}^{i-1}\big(f(V_j) + f(V-V_j)\big) - (i-2)f(V). \tag{5}$$

(Notice that the right hand side of (5) varies with the choice of the last member in $\mathcal{P}$.)

Proof. We proceed by induction on i. Clearly it holds for $i = 1$. Suppose that it holds for $i - 1$. Let us consider an i-partition $\mathcal{P} = \{V_1, V_2, \ldots, V_i\}$.

Since $\mathcal{P}_{i-1}$ is an $(i-1)$-partition, there must exist $W \in \mathcal{P}_{i-1}$ and $V_h, V_\ell \in \mathcal{P}$ with $h < \ell$ satisfying $W \cap V_h \neq \emptyset \neq W \cap V_\ell$. Hence the pair $(W \cap V_h, W)$ is a candidate for Line 3 in the $(i-1)^{th}$ iteration of GSA. Therefore we have

$$\begin{aligned} f(\mathcal{P}_i) - f(\mathcal{P}_{i-1}) \; &\leq \; f(W \cap V_h) + f(W - V_h) - f(W) \\ &\leq f(V_h) + f(W - V_h) - f(W \cup V_h) \; \leq \; f(V_h) + f(V - V_h) - f(V). \end{aligned} \tag{6}$$

(The last two inequalities are obtained by submodularity.)

By applying the induction hypothesis on $i-1$ to $\mathcal{P}_{i-1}$ and an $(i-1)$-partition $\mathcal{P}' \triangleq \{V_1, \ldots, V_{h-1}, V_{h+1}, \ldots, V_{i-1}, V_h \cup V_i\}$, we have

$$f(\mathcal{P}_{i-1}) \; \leq \sum_{1 \leq j \leq i-1, \; j \neq h} \big(f(V_j) + f(V - V_j)\big) - (i-3) f(V). \tag{7}$$

The induction is then complete by (6) and (7). □

Remark. The proof shows that Lemma 1 is valid if $f(X \cap Y) + f(X - Y) - f(X) \leq f(Y) + f(V - Y) - f(V)$ holds for all sets $X, Y \subseteq V$ satisfying $\emptyset \subset X \cap Y \subset X$, which is satisfied in a submodular system. □

Lemma 1 does not immediately provide the performance guarantee. Let us derive them for some MPPs in the next two subsections.

2.3. Performance analysis for MPP-NTs

Theorem 4. *Given a submodular system (V, f) and an integer $k \geq 2$, where $f(V) \geq 0$ holds, GSA finds a k-partition of V with weight at most $(1+\alpha)(1-\frac{1}{k})opt$, where opt is the weight of a minimum k-partition of V and α is any constant that satisfies $\sum_{i=1}^{k} f(V - V_i) \leq \alpha \sum_{i=1}^{k} f(V_i)$ for all k-partitions $\{V_1, \ldots, V_k\}$ of V.*

Proof. Let $\mathcal{P}^* = \{V_1^*, V_2^*, \ldots, V_k^*\}$ be a minimum k-partition of V such that $f(V_k^*) + f(V - V_k^*) = \max_{1 \leq i \leq k}\{f(V_i^*) + f(V - V_i^*)\}$ holds. Thus $f(V_k^*) + f(V - V_k^*) \geq \frac{1}{k}\sum_{i=1}^{k}(f(V_i^*) + f(V - V_i^*))$ holds. Hence we have

$$\begin{aligned} \sum_{i=1}^{k-1} \big(f(V_i^*) + f(V - V_i^*)\big) \; &\leq \; (1 - \frac{1}{k}) \sum_{i=1}^{k} (f(V_i^*) + f(V - V_i^*)) \\ &\leq \; (1 - \frac{1}{k})(1+\alpha) \sum_{i=1}^{k} f(V_i^*) \; = \; (1+\alpha)(1 - \frac{1}{k}) opt. \end{aligned}$$

On the other hand, by Lemma 1 (and $f(V) \geq 0$), GSA finds a k-partition with weight at most $\sum_{i=1}^{k-1}\big(f(V_i^*) + f(V - V_i^*)\big)$. The proof is then complete. □

Obviously we can choose $\alpha = 1$ for symmetric functions and obtain the next corollaries.

Corollary 1 (Queyranne [26]). *The k-PPSSS for a nonnegative symmetric submodular system (V, f) can be approximated within a factor of $2 - \frac{2}{k}$ in $O(kn^3\theta)$ time.* □

Corollary 2 (Saran and Vazirani [27], Kapoor [13]). *The k-PPG can be approximated within a factor of $2 - \frac{2}{k}$ in $O(kn(m + n \log n))$ time.* □

Let us consider an application of GSA. Given a graph G with a nonnegative weight function w on edges, the *strength* $\sigma(G, w)$ of G was introduced by Gusfield [10] and Cunningham [5] as a measure of network invulnerability, which is defined as

$$\sigma(G, w) = \min_{2\le k\le n}\left\{\frac{w(E_k)}{k-1} \;\middle|\; E_k \text{ is a minimum } k\text{-way cut of } G\right\}.$$

The strength of a graph can be found in $O(mn^2(m + n \log n))$ time ([5]). On the other hand, by applying GSA to n-PPG in G (which finds a near-minimum i-way cut for every $i = 2, \ldots, n$), we have the next result.

Corollary 3. *Given a graph G with a nonnegative weight function w on edges, a value σ satisfying $\sigma(G, w) \le \sigma \le 2\sigma(G, w)$ can be found in $O(n^2(m + n \log n))$ time.* □

Let us next show that k-PPSS can be approximated within a factor of $k - 1$ by GSA.

Lemma 2. *Let $\{V_1, \ldots, V_k\}$ be a k-partition in a submodular system (V, f). Then*

$$f(V_i) + f(V - V_i) \le \sum_{j=1}^{k} f(V_j) - (k-2)f(\emptyset),\; i = 1, 2, \ldots, k.$$

Proof. For any two disjoint sets $X, Y \subseteq V$, $f(X \cup Y) \le f(X) + f(Y) - f(\emptyset)$ holds by submodularity. An easy induction then shows that $f(V - V_i) = f(\bigcup_{j\ne i} V_j) \le \sum_{j\ne i} f(V_j) - (k-2)f(\emptyset)$, proving the lemma. □

Theorem 5. *Given a submodular system (V, f), where $f(V) + (k-1)f(\emptyset) \ge 0$ holds, MPP-NT can be approximated within a factor of $k - 1$ in $O(kn^3\theta)$ time.*

Proof. We first show that every k-partition $\mathcal{P} = \{V_1, V_2, \ldots, V_k\}$ has nonnegative weight (for the definition of performance guarantee). In fact, by Lemma 2, we have

$$\begin{aligned} f(\mathcal{P}) = \sum_{i=1}^{k} f(V_i) &\ge f(V_1) + f(V - V_1) + (k-2)f(\emptyset) \\ &\ge f(V) + (k-1)f(\emptyset) \ge 0. \end{aligned}$$

The theorem is then shown by combining Lemma 1 and Lemma 2. □

Corollary 4. *The k-PPSS can be approximated by GSA within factor $k - 1$ in $O(kn^3\theta)$ time.* □

Let us next derive the performance guarantee for k-PPMSS.

Theorem 6. *Given a submodular system (V, f), where $f(V) + f(\emptyset) \geq f(S)$ holds for any nonempty subset S of V, MPP-NT can be approximated by GSA within a factor of $2 - \frac{2}{k}$ in $O(kn^3\theta)$ time.*

Proof. We first show $f(S) \geq 0$, $\emptyset \subset S \subset V$. This is because $f(S) + f(V-S) \geq f(V) + f(\emptyset)$ holds by submodularity, and $f(V) + f(\emptyset) \geq f(V-S)$ holds by assumption. Notice that this fact implies that MPP-NT in (V, f) has nonnegative optimum.

Let $\mathcal{P}^* = \{V_1^*, V_2^*, \ldots, V_k^*\}$ be a minimum k-partition, where $f(V_{k-1}^*) + f(V_k^*) = \max_{1 \leq i < j \leq k}\{f(V_i^*) + f(V_j^*)\} \geq \frac{2}{k}\sum_{i=1}^k f(V_i^*)$ holds. By Lemma 1, we see that GSA finds a k-partition $\mathcal{P}_k$ with weight

$$f(\mathcal{P}_k) \leq \sum_{i=1}^{k-1}\left(f(V_i^*) + f(V - V_i^*)\right) - (k-2)f(V).$$

Notice that $f(V - V_i^*) \leq f(V) + f(\emptyset)$ holds by assumption. Hence we have

$$f(\mathcal{P}_k) \leq \sum_{i=1}^{k-1} f(V_i^*) + f(V - V_{k-1}^*) + (k-2)f(\emptyset) \leq \sum_{i=1}^{k-1} f(V_i^*) + \sum_{1 \leq i \leq k,\, i \neq k-1} f(V_i^*)$$

$$= 2\sum_{i=1}^{k} f(V_i^*) - f(V_{k-1}^*) - f(V_k^*) \leq (2 - \frac{2}{k})\sum_{i=1}^{k} f(V_i^*).$$

(The second inequality is by Lemma 2.) □

Corollary 5. *The k-PPMSS can be approximated by GSA within a factor of $2 - \frac{2}{k}$ in $O(kn^3\theta)$ time.* □

Our derivation of the guarantees is not only simple and unified, but also allows us to use approximation algorithms in Line 3 of GSA. Let us discuss this in the following.

Assume that $f(S) + f(W-S) - f(W) \geq 0$ holds for all $\emptyset \subset S \subset W \subseteq V$ (which is true if $f(\emptyset) \geq 0$). Suppose that a ρ-approximation algorithm is used in Line 3 of GSA. We observe that, by a straightforward induction as we used to prove Lemma 1, the i-partition $\mathcal{P}_i'$ $(1 \leq i \leq k)$ obtained by this variant of GSA has weight

$$f(\mathcal{P}_i') \leq \rho\left(\sum_{j=1}^{i-1}\left(f(V_j) + f(V - V_j)\right) - (i-2)f(V)\right)$$

for any k-partition $\mathcal{P} = \{V_1, \ldots, V_k\}$. Hence we have the next theorem.

Theorem 7. *The variant of GSA that uses a ρ-approximation algorithm in Line 3 is a $2\rho(1 - \frac{1}{k})$-approximation algorithm for k-PPSSS.* □

As a result, we obtain the next corollary by using the linear-time $(2 + \epsilon)$-approximation algorithm [20] for the minimum cut problem in graphs with unit edge weights.

Corollary 6. *The k-PPG with unit edge weight can be approximated within a factor of $(4 + \epsilon)(1 - \frac{1}{k})$ in $O(k(n + m))$ time, where $\epsilon \in (0, 1)$ is an arbitrary constant.* □

Obviously, similar approximation results can be obtained for other MPP-NTs.

2.4. Performance analysis for MPP-NTs in hypergraphs

Let us next consider the k-partition problems in hypergraphs. Let $H = (V, E)$ be a hypergraph with a hyperedge weight function $w : E \to \mathbf{R}^+$. Recall that three types of k-partition problems, k-PPH-T1, k-PPH-T2, and k-PPH-T3, employ objective functions $f_1 = \frac{w(E)}{k} - w_{\text{in}}$, $f_2 = w_{\text{ex}}$ and $f_3 = w_{\text{in}} + w_{\text{ex}} - \frac{w(E)}{k}$, respectively.

First of all, observe that GSA is a $(2 - \frac{2}{k})$-approximation algorithm for k-PPH-T2, since it is a special case of k-PPSSS. Let us consider the running time. Since w_{ex} can be evaluated in $O(D)$ time in a straightforward manner (recall that D is the sum of degrees of hyperedges), GSA can be implemented to run in $O(kn^3 D)$ time. In the next theorem, we show that a faster implementation is available.

Denote the set of hyperedges *between* two vertex subsets V_1 and V_2 by

$$E(V_1 : V_2) \triangleq \{e \in E \mid e \cap V_1 \neq \emptyset \neq e \cap V_2,\ e \subseteq V_1 \cup V_2\}.$$

Let $\delta(S) \triangleq E(S : V - S) = \{e \in E \mid \emptyset \subset e \cap S \subset e\}$ (note $w_{\text{ex}}(S) = w(\delta(S))$).

Theorem 8. *The k-PPH-T2 can be approximated by GSA within a factor of $2 - \frac{2}{k}$ in $O(kn(n \log n + D))$ time.*

Proof. The guarantee is implied by Corollary 1. We give a fast implementation.

We will show that, for any fixed $W \subseteq V$, a nonempty subset S of W minimizing $w_{\text{ex}}(S) + w_{\text{ex}}(W - S) - w_{\text{ex}}(W)$ can be found in $O(|W|^2 \log |W| + |W|D)$ time. This implies that Line 3 of GSA can be executed in $O(\sum_{W \in \mathcal{P}_i} (|W|^2 \log |W| + |W|D)) = O(n^2 \log n + nD)$ time, hence the claimed running time. Let us show how to minimize $w_{\text{ex}}(S) + w_{\text{ex}}(W - S) - w_{\text{ex}}(W)$. Since

$$\begin{aligned} w_{\text{ex}}(S) + w_{\text{ex}}(W - S) - w_{\text{ex}}(W) &= \sum_{e \in \delta(S)} w(e) + \sum_{e \in \delta(W-S)} w(e) - \sum_{e \in \delta(W)} w(e) \\ &= 2 \left(\sum_{e \in E(S:W-S)} w(e) + \sum_{e \in \delta(S) \cap \delta(W-S) \cap \delta(W)} \frac{w(e)}{2} \right), \end{aligned}$$

we see that the minimization of $w_{\text{ex}}(S) + w_{\text{ex}}(W - S) - w_{\text{ex}}(W)$ can be reduced to 2-PPH-T2 (i.e., the minimum cut problem) in hypergraph $H[W] \triangleq (W, E[W])$, where $E[W] \triangleq \{e \cap W | e \in E\} - \{\emptyset\}$, with weight function $w' : E[W] \to \mathbf{R}^+$ defined by

$$w'(e \cap W) = \begin{cases} w(e) & \text{if } e \subseteq W, \\ w(e)/2 & \text{if } e \in \delta(W). \end{cases} \tag{8}$$

(Notice that there may exist multiple hyperedges.) Using the algorithm in [16], we can minimize it in $O(|W|^2 \log |W| + |W|D)$ time. □

Next, let us consider k-PPH-T1 and k-PPH-T3. Functions $f_1 = \frac{w(E)}{k} - w_{\text{in}}$ and $f_3 = w_{\text{in}} + w_{\text{ex}} - \frac{w(E)}{k}$ are not nonnegative or symmetric in general. Hence we cannot apply Theorem 4 or Corollary 1. Nevertheless, since both Theorem 2 and Lemma 1 do not require nonnegative or symmetric functions, we can still use GSA to find a k-partition in polynomial time (by Theorem 2), and estimate the guarantees by Lemma 1. We obtain the next two theorems, where faster implementations are provided.

Theorem 9. *k-PPH-T1 can be approximated by GSA within factor* $\min\{k, d^+_{\max}\}(1-\frac{1}{k})$ *in* $O(kn(n\log n + D))$ *time, where* $d^+_{\max}$ *is the maximum degree of hyperedges of positive weights.*

Proof. We first derive the performance guarantee of GSA. Let $\mathcal{P}^* = \{V_1^*, V_2^*, \ldots, V_k^*\}$ be an optimal solution with $w_{\text{ex}}(V_k^*) = \max_{1\le i\le k}\{w_{\text{ex}}(V_i^*)\} \ge \frac{1}{k}\sum_{i=1}^{k} w_{\text{ex}}(V_i^*)$. By Lemma 1, GSA finds a k-partition whose weight is at most

$$\begin{aligned}
&\sum_{i=1}^{k-1}\big(f_1(V_i^*) + f_1(V - V_i^*)\big) - (k-2)f_1(V)\\
&\quad = \sum_{i=1}^{k-1}\left(\big(\frac{w(E)}{k} - w_{\text{in}}(V_i^*)\big) + \big(\frac{w(E)}{k} - w_{\text{in}}(V - V_i^*)\big)\right)\\
&\quad\quad -(k-2)\left(\frac{w(E)}{k} - w_{\text{in}}(V)\right)\\
&\quad = \sum_{i=1}^{k-1}\big(w(E) - w_{\text{in}}(V_i^*) - w_{\text{in}}(V - V_i^*)\big) \;=\; \sum_{i=1}^{k-1} w_{\text{ex}}(V_i^*)\\
&\quad \le (1-\frac{1}{k})\sum_{i=1}^{k} w_{\text{ex}}(V_i^*) \;\le\; (1-\frac{1}{k})\min\{k, d^+_{\max}\}\sum_{i=1}^{k} f_1(V_i^*).
\end{aligned}$$

The last inequality is based on the remark of Definition 1. Notice that we only need to consider hyperedges of positive weights, for which $p_e \le \min\{k, d^+_{\max}\}$ holds.

Since we can evaluate $f_1 = \frac{w(E)}{k} - w_{\text{in}}$ in $O(D)$ time, GSA for k-PPH-T1 can be executed in $O(kn^3D)$ time. We give a faster implementation in the following. Similarly to the proof of Theorem 8, we only need to show that a nonempty set $S \subset W$ minimizing $f_1(S) + f_1(W - S) - f_1(W)$ can be found in $O(|W|^2\log|W| + |W|D)$ time for any fixed $W \subseteq V$. For this, notice that

$$f_1(S) + f_1(W-S) - f_1(W) \;=\; w_{\text{in}}(W) - w_{\text{in}}(S) - w_{\text{in}}(W-S) + \frac{w(E)}{k}.$$

Thus the minimization of $f_1(S) + f_1(W-S) - f_1(W)$ can be reduced to 2-PPH-T1 (i.e., the minimum cut problem) in hypergraph $H_W \triangleq (W, E_W)$, where $E_W = E(W : W) = \{e \in E \mid e \subseteq W\}$, and hyperedges in E_W have the same weights as in H. Using the algorithm in [16], we can minimize it in $O(|W|^2\log|W| + |W|D)$ time. □

Theorem 10. *The k-PPH-T3 can be approximated by GSA within a factor of* $2 - \frac{2}{k}$ *in* $O(kn(n\log n + D))$ *time.*

Proof. We first derive the guarantee (notice that it is not a conclusion of Corollary 5). Let $\mathcal{P}^* = \{V_1^*, V_2^*, \ldots, V_k^*\}$ be an optimal solution with $w_{\text{ex}}(V_k^*) = \max w_{\text{ex}}(V_i^*) \ge \frac{1}{k}\sum_{i=1}^{k} w_{\text{ex}}(V_i^*)$. By Lemma 1, GSA finds a k-partition whose weight is at most

$$\sum_{i=1}^{k-1}\big(f_3(V_i^*)+f_3(V-V_i^*)\big)-(k-2)f_3(V)$$
$$=\sum_{i=1}^{k-1}\bigg(\Big(w_{\text{in}}(V_i^*)+w_{\text{ex}}(V_i^*)-\frac{w(E)}{k}\Big)+\Big(w_{\text{in}}(V-V_i^*)+w_{\text{ex}}(V-V_i^*)-\frac{w(E)}{k}\Big)\bigg)-(k-2)\Big(w_{\text{in}}(V)+w_{\text{ex}}(V)-\frac{w(E)}{k}\Big)$$
$$=\sum_{i=1}^{k-1}\Big(w_{\text{in}}(V_i^*)+w_{\text{ex}}(V_i^*)+w_{\text{in}}(V-V_i^*)+w_{\text{ex}}(V-V_i^*)-w(E)\Big)$$
$$=\sum_{i=1}^{k-1}w_{\text{ex}}(V_i^*)\ \le\ (1-\frac{1}{k})\sum_{i=1}^{k}w_{\text{ex}}(V_i^*)\ \le\ (2-\frac{2}{k})\sum_{i=1}^{k}f_3(V_i^*).$$

Again, the last inequality is based on the remark of Definition 1. Similarly as before, we next show how to minimize $f_3(S)+f_3(W-S)-f_3(W)$ in $O(|W|^2\log|W|+|W|D)$ time for any fixed $W\subseteq V$. For this, notice that

$$f_3(S)+f_3(W-S)-f_3(W)$$
$$=\Big(w_{\text{in}}(S)+w_{\text{ex}}(S)-\frac{w(E)}{k}\Big)+\Big(w_{\text{in}}(W-S)+w_{\text{ex}}(W-S)-\frac{w(E)}{k}\Big)-\Big(w_{\text{in}}(W)+w_{\text{ex}}(W)-\frac{w(E)}{k}\Big)$$
$$=\left(\sum_{e\in E(S:W-S)}w(e)\ +\sum_{e\in\delta(S)\cap\delta(W-S)\cap\delta(W)}w(e)\right)-\frac{w(E)}{k}.$$

Thus the minimization of $f_3(S)+f_3(W-S)-f_3(W)$ can be reduced to 2-PPH-T3 (i.e., the minimum cut problem) in hypergraph $H[W]=(W,E[W])$, in which each hyperedge has the same weight as in H. Again, we can use the algorithm in [16]. □

3. Modified GSA (M-GSA) for general MPP

3.1. Algorithm description and main lemma

We introduce in this section a slightly modified version of GSA for the general MPP problems. Actually, by considering only k-partitions of target-split type, we can extend GSA in a straightforward manner. We list the resulting algorithm M-GSA in Fig. 2.

Notice that only Line 3 is different from GSA, where we want to minimize $f(S)+f(W-S)-f(W)$ under an additional constraint: $S\cap T\neq\emptyset\neq(W-S)\cap T$. This ensures the feasibility of the output solution $\mathcal{P}_k$. Let us first consider this minimization. If $W\subseteq T$, the constraints reduce to $\emptyset\subset S\subset W$ only, hence the minimization can be achieved in $O(|W|^3\theta)$ time by Theorem 2. Otherwise, as will be shown in Theorem 11, it can be achieved by solving the next *partial s,t-partition problem.*

Input: A submodular system (V, f), a target set $T \subseteq V$ and an integer k satisfying $1 \leq k \leq |T|$.
Output: A target-split $\mathcal{P}_k$ of V.

1 $\mathcal{P}_1 \leftarrow \{V\}$
2 **for** $i = 1, \ldots, k-1$ **do**
3 $\quad (S_i, W_i) \leftarrow \text{argmin}\{f(S) + f(W-S) - f(W) \mid \emptyset \subset S \subset W,\ W \in \mathcal{P}_i,$ $S \cap T \neq \emptyset \neq (W-S) \cap T\}$
4 $\quad \mathcal{P}_{i+1} \leftarrow (\mathcal{P}_i - \{W_i\}) \cup \{S_i, W_i - S_i\}$
5 **end** /* for */

Fig. 2. Modified greedy splitting algorithm (M-GSA) for MPP.

Problem 5 (Partial *s*,*t*-partition problem). Given a submodular system (V, f) with a subset W of V and distinct $s, t \in W$, find a nonempty proper subset S of W such that

$$\begin{aligned} &\text{minimize} && f(S) + f(W-S) \\ &\text{subject to} && s \in S,\ t \in W-S. \end{aligned}$$

Lemma 3. *The partial s,t-partition problem can be solved in* $O(\theta|W|^7 \log |W|)$ *time.*

Proof. Let $g(S) = f(S \cup \{s\}) + f(W - (S \cup \{s\}))$, $S \subseteq W - \{s, t\}$. Consider system $(W - \{s, t\}, g)$. Obviously, we only need to find an $S' \subseteq W - \{s, t\}$ to minimizes g, since $S^* = S' \cup \{s\}$ will be an optimal solution of the partial s,t-partition problem. It is easy to verify that g is submodular. An algorithm in [12] for minimizing submodular functions has a running time of $O(\theta|W|^7 \log |W|)$, proving the lemma. □

Theorem 11. *Given a submodular system* (V, f)*,* $T \subseteq V$ *and an integer* $k \geq 1$*, M-GSA has* $O(k\theta|T|n^7 \log n)$ *running time.*

Proof. For a fixed $W \subseteq V$, the minimization of $f(S) + f(W-S)$ with $\emptyset \subset S \subset W$ and $S \cap T \neq \emptyset \neq (W-S) \cap T$ can be achieved by applying at most $|T \cap W| - 1$ partial s, t-partition minimizations in the following manner. Choose an arbitrary $s \in W \cap T$. Compute a minimum partial s, t-partition for every $t \in (W \cap T - \{s\})$. Obviously the minimum minimum-partial-s,t-partition among all t is a desired solution. Hence Line 3 of M-GSA can be carried out in $\sum_{W \in \mathcal{P}_i} O(\theta|T \cap W||W|^7 \log |W|) = O(\theta|T|n^7 \log n)$ time, implying the theorem. □

Remark. Of course we may have faster implementations in special cases. For instance, it is easy to see that, the partial s,t-partition problem for k-TPG reduces to the minimum s, t-cut problem (i.e. 2-MCP) which can be solved by a single maxflow computation. Thus the running time of M-GSA for k-TPG is $O(k|T|mn \log(n^2/m))$ by using the $O(mn \log(n^2/m))$ time maxflow algorithm [9]. □

Let us next consider the performance guarantees of M-GSA. We first give a technical lemma which is analogous to Lemma 1.

Lemma 4. *Let* $\mathcal{P}_i$ *be the i-target-split found by M-GSA in the* $(i-1)^{th}$ *iteration,* $1 \leq i \leq k$*. For any i-target-split* $\mathcal{P} = \{V_1, V_2, \ldots, V_i\}$*, it holds that*

$$f(\mathcal{P}_i) \leq \sum_{j=1}^{i-1} \big(f(V_j) + f(V - V_j)\big) - (i-2) f(V). \tag{9}$$

Proof. Analogous to the proof of Lemma 1. □

3.2. Performance analysis

Analogously to the proof of Theorem 5, we have the next results by Theorem 11, Lemma 2 and Lemma 4.

Theorem 12. *Given a system (V, T, k, f), where $f(V)+(k-1)f(\emptyset) \geq 0$ holds, MPP can be approximated by M-GSA within a factor of $k-1$ in $O(k\theta|T|n^7 \log n)$ time.* □

Corollary 7. *Problem k-TPSS can be approximated by M-GSA within factor $k-1$ in $O(k\theta|T|n^7 \log n)$ time.* □

Analogously to Corollary 1, the next result can be shown by Theorem 11 and Lemma 4.

Theorem 13. *The k-TPSSS can be approximated by M-GSA within a factor of $2-\frac{2}{k}$ in $O(k\theta|T|n^7 \log n)$ time.* □

We have the next corollary by combining Theorem 13 and the remark of Theorem 11.

Corollary 8 (Maeda, Nagamochi and Ibaraki [21]). *The k-TPG can be approximated by M-GSA within a factor of $2-\frac{2}{k}$ in $O(k|T|mn \log(n^2/m))$ time.* □

The next two results can be shown analogously to the proofs of Theorem 6 and Corollary 5, respectively.

Theorem 14. *Given a system (V, T, k, f), where $f(V)+f(\emptyset) \geq f(S)$ holds for any nonempty subset S of V, MPP can be approximated by M-GSA within a factor of $2-\frac{2}{k}$ in $O(k\theta|T|n^7 \log n)$ time.* □

Corollary 9. *The k-TPMSS can be approximated by M-GSA within a factor of $2-\frac{2}{k}$ in $O(k\theta|T|n^7 \log n)$ time.* □

Theorem 15. *The k-TPH-T2 can be approximated by M-GSA within a factor of $2-\frac{2}{k}$ in $O(k|T|m'n' \log(n'^2/m'))$ time, where $m' = 2D+m$, $n' = n+m$.*

Proof. The guarantee is implied by Theorem 13. We show a fast implementation.

Similarly as before, we see that, the partial s,t-partition problem for k-TPH-T2 reduces to 2-TPH-T2 with target set $\{s, t\}$ in hypergraph $H[W] = (W, E[W])$ with a modified weight function w' (see (8)). It is known that the resulting problem can be solved by a single maxflow computation ([17]). A calculation shows the theorem. □

Analogously to the proof of Theorems 15, 9 and 10, we have the next two theorems.

Theorem 16. *Problem k-TPH-T1 can be approximated by M-GSA within a factor of $\min\{k, d^+_{\max}\}(1-\frac{1}{k})$ in $O(k|T|m'n' \log(n'^2/m'))$ time, where $m' = 2D+m$, $n' = n+m$.* □

Theorem 17. *The k-TPH-T3 can be approximated by M-GSA within factor $2-\frac{2}{k}$ in $O(k|T|m'n' \log(n'^2/m')$ time, where $m' = 2D+m$, $n' = n+m$.* □

Again, our proof allows us to use approximate algorithms in Line 3 of M-GSA. Assume that $f(S) + f(W - S) - f(W) \geq 0$ holds for $\emptyset \subset S \subset W$, $W \in \mathcal{P}_i$ and $S \cap T \neq \emptyset \neq (W - S) \cap T$ (which is true if $f(\emptyset) \geq 0$ holds). We have the next theorem.

Theorem 18. *The variant of M-GSA that uses a ρ-approximation algorithm in Line* 3 *is a $\rho(k-1)$-approximation (resp., $2\rho(1-\frac{1}{k})$-approximation) algorithm for k-PPSS (resp., k-TPSSS).* □

It is easy to see that similar results are available for other MPPs.

4. Tight example

In this section, we construct tight examples for GSA (clearly such examples are also tight examples for M-GSA). For this, we need an easy preliminary property, whose proof can be done in a straightforward manner and hence is omitted.

Lemma 5. *Let (V, f) and (W, g) be two submodular systems, where V and W are not necessarily disjoint. Define function $h : 2^{V \cup W} \rightarrow \mathbf{R}$ by $h(S) = f(S \cap V) + g(S \cap W)$, $S \subseteq V \cup W$. Then $(V \cup W, h)$ is a submodular system.* □

Theorem 19. *For any $\epsilon > 0$, there exists a nonnegative submodular system for which GSA always finds a k-partition whose weight is at least $k - 1 - \epsilon$ times of the optimum.*

Proof. Let $V = \{v_0, v_1, \ldots, v_{k-1}\}$, and let $f : 2^V \rightarrow \mathbf{R}^+$ be defined by $f(S) = |S|$ if $S \subset V$ and $f(V) = 0$. Let $W = \{v_0, w_1, \ldots, w_{k-1}\}$, $V \cap W = \{v_0\}$. Let g be the cut function of graph (W, E) whose edges in $E = \{\{v_0, w_1\}, \{v_0, w_2\}, \ldots, \{v_0, w_{k-1}\}\}$ are weighted $\frac{k-\epsilon}{2}$ (see Fig. 3). Obviously f and g are submodular.

Let us consider a submodular system $(V \cup W, h)$, where h is defined by $h(S) = f(S \cap V) + g(S \cap W)$, $S \subseteq V \cup W$. There is a k-partition (the optimal solution) $\mathcal{P} = \{\{v_1\}, \{v_2\}, \ldots, \{v_{k-1}\}, W\}$ whose weight is k. On the other hand, since any 2-partition separating some pair of members in V has weight at least k, the minimum 2-partition of $(V \cup W, h)$ is $\{\{w_i\}, (V \cup W) - \{w_i\}\}$, $1 \leq i \leq k$ with weight $k - \epsilon$. Thus GSA first finds a 2-partition $\{\{w_i\}, (V \cup W) - \{w_i\}\}$ for some i. And finally it will output the k-partition $\mathcal{P}_k = \{\{w_1\}, \{w_2\}, \ldots, \{w_{k-1}\}, V\}$, whose weight is $(k-1)(k-\epsilon)$. Thus the guarantee is at least as bad as $\frac{(k-1)(k-\epsilon)}{k} > k - 1 - \epsilon$. □

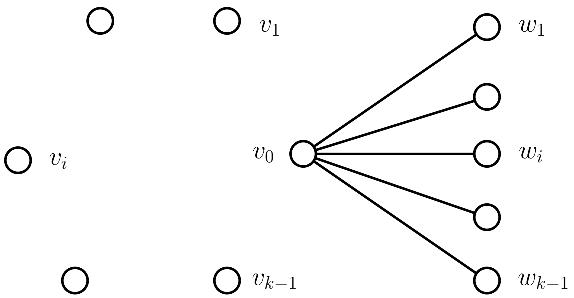

Fig. 3. A tight example for GSA applied to k-PPSS.

A tight example for k-PPG is known, see [27]. Notice that it also serves as a tight example for k-PPH-Ti, $i = 1, 2, 3$, and k-PPSSS.

5. Conclusion and remark

In this paper, we have presented a simple and unified approach for developing and analyzing approximation algorithms for various multiway partition problems (MPPs). The main idea is to construct a near-optimal solution by greedily increasing the size of partition by one in each iteration. Various approximation results are shown.

Naturally, one may consider to greedily increase the size of partition by two in one iteration. Actually we can show that improved guarantees can be achieved for a number of problem classes (see [31, 32]). On the other hand, however, polynomial time implementation for the general MPP-NT is still open (which would be available if the 3-PPSS was solvable in polynomial time), whereas it seems not available for the general MPP unless $P = NP$.

Finally, we note that our approach (actually the main lemmas) cannot be extended to analyze the greedy algorithm that increases the size of partition by three (or more) in each iteration, see [31]. Further details are available in the thesis [30].

Acknowledgements. This research is partially supported by the Scientific Grant-in-Aid from Ministry of Education, Science, Sports and Culture of Japan. The authors would like to thank the anonymous referees for their valuable comments and suggestions.

References

1. Benczúr, A.A.: Counterexamples for directed and node capacitated cut-trees. SIAM J. Comput. **24**, 505–510 (1995)
2. Burlet, M., Goldschmidt, O.: A new and improved algorithm for the 3-cut problem. Oper. Res. Lett. **21**, 225–227 (1997)
3. Călinescu, G., Karloff, H., Rabani, Y.: An improved approximation algorithm for multiway cut. J. Comput. System Sci. **60**, 564–574 (2000)
4. Chopra, S., Owen, J.H.: A note on formulations for the A-partition problem on hypergraphs. Discrete Appl. Math. **90**, 115–133 (1999)
5. Cunningham, W.H.: Optimal attack and reinforcement of a network. J. Assoc. Comput. Mach. **32**, 549–561 (1985)
6. Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D, Yannakakis, M.: The complexity of multiterminal cuts. SIAM J. Comput. **23**, 864–894 (1994)
7. Gomory, R.E., Hu, T.C.: Multi-terminal network flows. J. Soc. Indust. Appl. Math. **9**, 551–570 (1961)
8. Goldschmidt, O., Hochbaum, D.S.: A polynomial algorithm for the k-cut problem for fixed k. Math. Oper. Res. **1,9** 24–37 (1994)
9. Goldberg, A.V., Tarjan, R.E.: A new approach to the maximum flow problem. J. Assoc. Comput. Mach. **35**, 921–940 (1988)
10. Gusfield, D.: Connectivity and edge-disjoint spanning trees. Inform. Process. Lett. **16**, 87–89 (1983)
11. Garg, N., Vazirani, V.V., Yannakakis, M.: Multiway cuts in directed and node weighted graphs (extended abstract). In Proc. ICALP 1994, LNCS **820**, 487–498 (1994)
12. Iwata, S., Fleischer, L.L., Fujishige, S.: A combinatorial strongly polynomial time algorithm for minimizing submodular functions. J. ACM **48**, 761–777 (2001)
13. Kapoor, S.: On minimum 3-cuts and approximating k-cuts using cut trees. In Proc. IPCO 1996, LNCS **1084**, 132–146 (1996)
14. Karger, D.R., Klein, P., Stein, C., Thorup, M., Young, N.E.: Rounding algorithms for a geometric embedding of minimum multiway cut. In Proc. STOC 1999, 668–678 (1999)

15. Korte, B., Vygen, J. : Combinatorial Optimization. Theory and Algorithms. Springer, Berlin (2000)
16. Klimmek, R., Wagner, F.: A simple hypergraph min cut algorithm. Technical Report B 96-02, Freie Universität Berlin (1996)
17. Lawler, E.L.: Cutsets and partitions of hypergraphs. Networks **3**, 275–285 (1973)
18. Lengauer, T.: Combinatorial Algorithms for Integrated Circuit Layout. Wiley, New York (1990)
19. Lee, C.H., Kim, M., Park, C.I. : An efficient k-way graph partitioning algorithm for task allocation in parallel computing systems. In Proc. IEEE Int. Conf. on Computer-Aided Design 1990, 748–751 (1990)
20. Matula, D.W.: A linear time $2 + \epsilon$ approximation algorithm for edge connectivity. In Proc. SODA 1993, 500–504 (1993)
21. Maeda, N., Nagamochi, H., Ibaraki, T.: Approximate algorithms for multiway objective point split problems of graphs (in Japanese). Computing Devices and Algorithms (in Japanese) (Kyoto, 1993). Surikaisekikenkyusho Kokyuroku **833**, 98–109 (1993)
22. Nagamochi, H., Ibaraki, T.: Computing edge connectivity in multigraphs and capacitated graphs. SIAM J. Discrete Math. **5**, 54–66 (1992)
23. Narayanan, H., Roy, S., Patkar, S.: Approximation algorithms for min-k-overlap problems using the principal lattice of partitions approach. J. Algorithms **21**, 306–330 (1996)
24. Pulleyblank, W.R.: Presentation at SIAM Meeting on Optimization, MIT, Boston (1992)
25. Queyranne, M.: Minimizing symmetric submodular functions. Math. Program. B **82**, 3–12 (1995)
26. Queyranne, M.: On optimum size-constrained set partitions. AUSSOIS 1999, France (1999)
27. Saran, H., Vazirani, V.V.: Finding k-cuts within twice the optimal. SIAM J. Comput. **24**, 101–108 (1995)
28. Tittmann, P.: Partitions and network reliability. Discrete Appl. Math. **95**, 445–453 (1999)
29. Vazirani, V.V.: Approximation Algorithms. Springer-Verlag, Berlin (2001)
30. Zhao, L.: Approximation algorithms for partition and design problems in networks. PhD Thesis, Graduate school of Informatics, Kyoto University, Japan (2002)
31. Zhao, L., Nagamochi, H., Ibaraki, T.: Approximating the minimum k-way cut in a graph via minimum 3-way cuts. J. Comb. Optim. **5**, 397–410 (2001)
32. Zhao, L., Nagamochi, H., Ibaraki, T.: On generalized greedy splitting algorithms for multiway partition problems. Discrete Appl. Math. (to appear). A preliminary version appeared in Proc. ISAAC 2001 (A unified framework for approximating multiway partition problems (extended abstract). LNCS **2223**, 682–694) (2004)