数理解析研究所講究録 1241 巻 2001 年 139-147

Greedy Splitting: A Unified Approach for Approximating Some Partition Problems

Liang Zhao †, Hiroshi Nagamochi ‡, and Toshihide Ibaraki †

† Dept. of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University Kyoto 606-8501, Japan

‡ Dept. of Information and Computer Sciences, Toyohashi University of Technology Toyohashi 441-8580, Japan

趙 亮 †, 永持 仁 ‡, 茨木 俊秀 †

† 京都大学情報学研究科数理工学専攻 〒 606-8501 京都府京都市左京区吉田本町

‡ 豊橋技術科学大学情報工学系 〒 441-8580 愛知県豊橋市天伯町雲雀ヶ丘 1-1

Abstract. We present a simple and unified framework for developing and analyzing approximation algorithms for some multiway partition problems (with or without terminals), including the *k-way cut* (or *k-cut*), *multiterminal cut* (or *multiway cut*), *hypergraph partition* and *target split.*

1 Introduction

Let V and $f: 2^V \rightarrow \mathbf{R}$ be a finite set and a set function respectively. Function f is *submodular* if $f(A)+f(B) \geq f(A\cap B)+f(A\cup B)$ for all subsets A and B of V. It is *symmetric* if $f(S)=f(V-S)$ for all $S\subseteq V$. A family $\mathcal{P}=\{V_1,\ldots,V_k\}$ of pairwise disjoint nonempty subsets of V whose union is V is called a *k-partition* of V. The cost of $\mathcal{P}$ (with respect to f) is defined as $f(\mathcal{P})=\sum_{i=1}^{k} f(V_i)$. Given a submodular system (V,f) where f is nonnegative, the *k-partition problem in submodular systems* (k-PPSS) is to find a k-partition of V with the minimum cost ($2\leq k\leq |V|-1$). The *k-partition problem in symmetric submodular systems* (k-PP3S) is the k-PPSS with symmetric f. We assume that f is given by an oracle that computes $f(S)$ in at most θ time for any $S\subseteq V$.

The *k-partition problem in hypergraphs* (k-PPH) is a special case of the k-PP3S in which V and f are respectively the vertex set and the *cut function* of a hypergraph with nonnegative hyperedge costs (i.e., for any $S\subseteq V$, $f(S)$ is the sum of costs of hyperedges that have at least one but not all endpoints in S. It is nonnegative, symmetric and can be easily seen to be submodular).

Our study starts from the *k-partition problem in graphs* (k-PPG). Given an undirected graph with nonnegative edge costs, the k-PPG asks to find a minimum cost edge subset whose removal leaves the graph with at least k connected components. The problem is also called the *k-way cut* or *k-cut problem.* Goldschmidt and Hochbaum [4] have shown that the k-PPG is NP-hard for arbitrary k even for unit edge costs, while it is solvable for any fixed k in $O(n^{k^2})$ time, where n is the number of vertices. Faster algorithms can be found in [8, 12, 13]. Saran and Vazirani [18] and Kapoor [6] showed that the k-PPG problem (for arbitrary k) can be approximated within factor $2-\frac{2}{k}$ in polynomial time. Recently, the authors [19] have given an approximation algorithm with improved performance guarantee about $2-\frac{3}{k}$.

Clearly the inclusion among the above problem classes is k-PPG $\subset$ k-PPH $\subset$ k-PP3S $\subset$ k-PPSS. Hence all of these are also NP-hard for arbitrary k. Queyranne [16] has shown that for any fixed k the k-PP3S is solvable in $O(|V|^{k^2}\theta)$ time. A faster algorithm for the 3-PP3S can be found in [12]. On the other hand, Queyranne [16] extends a greedy algorithm in [6, 18] to show that the k-PP3S can be approximated within factor $2-\frac{2}{k}$ in polynomial time. We note that his proof and the proofs of [6, 18] all use lower bounds derived from the so-called cut tree (or Gomory-Hu tree) for f (or for undirected graph), and are rather complicated and work only for symmetric submodular systems. As will be seen in the following, our approach in this paper works for any submodular system and gives a much simpler proof to show the same results.

We first show that the 2-PPSS is solvable in $O(|V|^3\theta)$ time, while we leave it open whether the k-PPSS can be solved in polynomial time for fixed $k\geq 3$. We then extend the greedy algorithm in [6, 16, 18] to the k-PPSS. It finds a k-partition of V by greedily "splitting" V via minimum 2-partition computations. We will give a simple proof to show that the performance guarantee is no worse than $(1+\alpha)(1-\frac{1}{k})$, where α is any number 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. We show that in general we can let $\alpha=k-1$, which implies the performance guarantee $k-1$. This is the first approximation algorithm for the k-PPSS. Furthermore, it is clear that we can let $\alpha = 1$ if f is symmetric, which implies the results of [6, 16, 18]. Several more applications will also be given.

We next consider to approximate the k-PPSS via minimum 2,3-partition computations. We will show some properties on the performance and use them to approximate the k-PPH by factor about $2 - \frac{3}{k}$. This extends our result [19] for the k-PPG and improves the previous best bound $2 - \frac{2}{k}$ (implied by the result for the k-PP3S due to Queyranne [16]).

Finally we extend our results to the *target split problem in submodular systems* (TSPSS), which for an additional given *target set* $T \subseteq V$ ($|T| \geq k$) asks to find a minimum k-partition $\{V_1, V_2, \ldots, V_k\}$ such that each V_i contains at least one target in T. A special case in which $|T| = k$, V and f are respectively the vertex set and the cut function of graphs is called the *multiterminal* (or previously *multiway*) *cut problem*, which is NP-hard even for $k = 3$ [3], and can be approximated within factor $1.5 - \frac{1}{k}$ [1], 1.3438 [7]. Clearly the TSPSS is a generalization of the k-PPSS and the multiterminal cut problem. We note that Maeda, Nagamochi and Ibaraki [11] have considered the target split problem in graphs and shown that it can be approximated within factor $2 - \frac{2}{k}$ in polynomial time. Our result will also give a simpler proof to show their result.

2 k-PPSS and Greedy Splitting Algorithm

We first observe that the 2-PPSS is solvable in polynomial time.

Theorem 1 (Queyranne [15]) *Given a symmetric submodular function $g: 2^V \to \mathbf{R}$, a nonempty proper subset S^* of V ($|V| \geq 2$) such that $g(S^*)$ is minimum can be found in $O(|V|^3\theta_g)$ time where θ_g is the time bound of the oracle for g.* □

Theorem 2 *Given a submodular function $f: 2^V \to \mathbf{R}$ and a $W \subseteq V$ ($|W| \geq 2$), 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.*

Proof. Let $g: 2^W \to \mathbf{R}$ be defined by $g(S) = f(S) + f(W - S)$ for all $S \subseteq W$. Notice that g is symmetric, submodular and for any $S \subseteq W$ we can compute $g(S)$ in at most 2θ time. Theorem 1 shows that such an S^* can be found in $O(|W|^3\theta)$ time. □

We next present a *greedy splitting* algorithm (GSA) for the k-PPSS in Table 1.

Table 1: Greedy splitting algorithm (GSA) for the k-PPSS.

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

GSA contains $k-1$ rounds and the i-th round computes an $(i+1)$-partition $\mathcal{P}_{i+1}$ of V, where $\mathcal{P}_1 = \{V\}$ and $\mathcal{P}_{i+1}$ is obtained by greedily "splitting" some member in $\mathcal{P}_i$ into two nonempty parts at the minimum cost. Formally, in the i-th round we compute a pair (S_i, W_i) that minimizes $f(S) + f(W - S) - f(W)$ (called the *splitting cost*) over all S and W such that $\emptyset \neq S \subset W$ and $W \in \mathcal{P}_i$. We get $\mathcal{P}_{i+1}$ from $\mathcal{P}_i$ by replacing W_i with S_i and $W_i - S_i$. Thus, for $\ell = 1, 2, \ldots, k$, it holds

$$f(\mathcal{P}_\ell) = f(V) + \sum_{i=1}^{\ell-1} (f(S_i) + f(W_i - S_i) - f(W_i)). \tag{1}$$

Clearly the output $\mathcal{P}_k$ of GSA is a k-partition of V. For any (fixed) $W \subseteq V$, Theorem 2 shows that we can find in $O(|W|^3\theta)$ time a nonempty proper subset S^* of W such that $f(S^*) + f(W - S^*)$ (hence $f(S^*) + f(W - S^*) - f(W)$) is minimum. Thus we can execute step 3 in $\sum_{W \in \mathcal{P}_i} O(|W|^3\theta) = O(|V|^3\theta)$ time. Hence the running time of GSA is $O(k|V|^3\theta)$.

To analyze the performance guarantee, we first go through a technical lemma.

Lemma 1 *For an $\ell \in \{1,\ldots,k\}$, let $\mathcal{P}_\ell$ be the ℓ-partition of V found by GSA in the $(\ell-1)$-th round. Then for any ordered ℓ-partition $\{V_1, V_2, \ldots, V_\ell\}$ of V, it holds that*

$$f(\mathcal{P}_\ell) \;\le\; \sum_{i=1}^{\ell-1}(f(V_i)+f(V-V_i)) - (\ell-2)f(V). \tag{2}$$

Proof. We proceed by induction on ℓ. It is trivial for $\ell = 1$. Suppose that it holds for $\ell - 1$. Consider an ordered ℓ-partition $\mathcal{P} = \{V_1, V_2, \ldots, V_\ell\}$ of V. Since $\mathcal{P}_{\ell-1}$ is an $(\ell-1)$-partition of V, there exist a $W \in \mathcal{P}_{\ell-1}$ and two distinct $V_j, V_h \in \mathcal{P}$ with $j < h$ such that $W \cap V_j \neq \emptyset \neq W \cap V_h$. We here consider the ordered $(\ell-1)$-partition $\mathcal{P}' = \{V_1, \ldots, V_{j-1}, V_{j+1}, \ldots, V_{\ell-1}, V_j \cup V_\ell\}$ where the order is the same as $\mathcal{P}$ except for that V_j is merged with the last member V_ℓ (notice $j < \ell$). By the induction hypothesis on $\ell - 1$, (2) holds for $\mathcal{P}_{\ell-1}$ and $\mathcal{P}'$, i.e.,

$$f(\mathcal{P}_{\ell-1}) \;\le\; \sum_{1\le i\le \ell-1,\; i\neq j} (f(V_i)+f(V-V_i) - (\ell-3)f(V). \tag{3}$$

Thus by (3) it suffices to show that

$$f(\mathcal{P}_\ell) - f(\mathcal{P}_{\ell-1}) \;\le\; f(V_j) + f(V-V_j) - f(V). \tag{4}$$

Notice that $W \cap V_j$ is a nonempty proper subset of W. Thus $(W \cap V_j, W)$ is a candidate for step 3 of GSA. Hence by the optimality of $(S_{\ell-1}, W_{\ell-1})$,

$$f(S_{\ell-1}) + f(W_{\ell-1} - S_{\ell-1}) - f(W_{\ell-1}) \;\le\; f(W \cap V_j) + f(W - V_j) - f(W). \tag{5}$$

The submodularity of f implies that the right hand of (5) is at most

$$f(V_j) + f(W - V_j) - f(W \cup V_j) \;\le\; f(V_j) + f(V - V_j) - f(V),$$

proving (4). □

Theorem 3 *Given a nonnegative submodular system (V, f), GSA finds a k-partition of V of cost at most $(1+\alpha)(1-\frac{1}{k})$ times the optimum, where α is any number that satisfies $\sum_{i=1}^{k} f(V-V_i) \le \alpha \sum_{i=1}^{k} f(V_i)$ for all k-partitions $\{V_1, \ldots, V_k\}$ of V.*

Proof. Let $\mathcal{P}^* = \{V_1^*, V_2^*, \ldots, V_k^*\}$ be an optimal k-partition of V with the order such that $f(V_k^*) + f(V - V_k^*) = \max_{1\le i\le k}\{f(V_i^*) + f(V - V_i^*)\}$. Then

$$\sum_{i=1}^{k-1}(f(V_i^*)+f(V-V_i^*)) \le (1-\frac{1}{k})\sum_{i=1}^{k}(f(V_i^*)+f(V-V_i^*)) \le (1+\alpha)(1-\frac{1}{k})\sum_{i=1}^{k} f(V_i^*).$$

On the other hand, by Lemma 1 GSA finds a k-partition of cost at most $\sum_{i=1}^{k-1}(f(V_i^*) + f(V - V_i^*))$ (note $f(V) \ge 0$). Hence the proof goes because $\sum_{i=1}^{k} f(V_i^*)$ is the optimum. □

For symmetric f, we can let $\alpha = 1$ and obtain the following corollaries.

Corollary 1 (Queyranne [16]) *The k-PP3S can be approximated within factor $2 - \frac{2}{k}$ in polynomial time.* □

Corollary 2 (Saran and Vazirani [18], Kapoor [6]) *The k-PPG problem can be approximated within factor $2 - \frac{2}{k}$ in polynomial time.* □

In general we cannot let $\alpha = 1$. Nevertheless, we show that $\alpha = k - 1$ is enough.

Lemma 2 $\sum_{i=1}^{k} f(V - V_i) \le (k-1)\sum_{i=1}^{k} f(V_i) - k(k-2)f(\emptyset)$ *holds for any k-partition $\{V_1, \ldots, V_k\}$ of a submodular system (V, f).*

Proof. For any two disjoint $A, B \subseteq V$, $f(A \cup B) \leq f(A) + f(B) - f(\emptyset)$ holds by the submodularity of f. Thus $f(V - V_i) = f(\bigcup_{j \neq i} V_j) \leq \sum_{j \neq i} f(V_j) - (k-2)f(\emptyset)$ for $i = 1, \ldots, k$. Hence the lemma goes. □

Notice that $f(\emptyset) \geq 0$ in the k-PPSS, which implies that $\alpha = k - 1$ is enough. Thus the performance guarantee of GSA for the k-PPSS is no worse than $k - 1$. We remark that the bound is also tight (a tight example will be given in the full paper). We summarize the arguments so far in the next theorem.

Theorem 4 *The k-PPSS can be approximated within factor $k-1$ in $O(k|V|^3\theta)$ time for any nonnegative submodular system (V, f), where θ is the time bound of the oracle for f.* □

Our proof is not only very simple but also allows us to plug some approximate algorithms into GSA. Suppose that a ρ-approximation algorithm for 2-PPSS is used. It is easy to see that the cost of the obtained k-partition is bounded by $\rho(1 + \alpha)(1 - \frac{1}{k})$ times the optimum.

Theorem 5 *The variant of GSA that uses a ρ-approximation algorithm for 2-PPSS to compute 2-partitions is a $\rho(1 + \alpha)(1 - \frac{1}{k})$ approximation algorithm for the k-PPSS, where α is any number 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.* □

As a result, we obtain the next corollary by using the linear time $(2 + \epsilon)$-approximation algorithm [10] for minimum cut problem in graphs with unit edge costs, where $\epsilon \in (0, 1)$ is an arbitrary number.

Corollary 3 *The k-PPG in graphs with unit edge costs can be approximated within factor $(4 + \epsilon)(1 - \frac{1}{k})$ in $O(k(n + m))$ time, where $\epsilon \in (0, 1)$ is a fixed number, and n and m are the numbers of vertices and edges respectively.* □

Before closing this section, we show important applications of our results to two variants of the k-PPH that arise from VLSI design [2, 9]. Let $H = (V, E)$ be a hypergraph with vertex set V and hyperedge set E. Let $c: E \to \mathbf{R}^+$ be a nonnegative hyperedge cost function. For a k-partition $\mathcal{P}$ of V, two types of cost to be minimized, $cost_1(\mathcal{P})$ and $cost_2(\mathcal{P})$, are introduced: $cost_1(\mathcal{P})$ counts the cost $c(e)$ of each hyperedge e $p - 1$ *times* if its endpoints of e belong to p distinct members in $\mathcal{P}$, while $cost_2(\mathcal{P})$ counts $c(e)$ *once* if its endpoints e belong to at least two distinct members in $\mathcal{P}$.

For the k-PPH with cost functions $cost_1$ and $cost_2$, the previous best approximation guarantees are $2 - \frac{2}{n}$ and $d_{\max}(1 - \frac{1}{n})$ respectively [14], where $d_{\max}$ is the maximum degree of hyperedges. We here show that better guarantees can be obtained by a simpler proof than [14]. For this, we define three set functions f_{ex}, f_{in} and $f: 2^V \to \mathbf{R}^+$ as follows. Let f_{ex} be the cut function of H. For any $S \subseteq V$, let $f_{\text{in}}(S)$ be the sum of costs of hyperedges whose endpoints are all in S, and $f(S) = f_{\text{ex}}(S) + f_{\text{in}}(S)$. Observe that the k-PPH with cost function $cost_1$ asks to find a k-partition $\mathcal{P} = \{V_1, V_2, \ldots, V_k\}$ of V that minimizes $\sum_{i=1}^{k} f(V_i) - f(V) = \sum_{i=1}^{k}(f(V_i) - \frac{f(V)}{k})$, while the k-PPH with cost function $cost_2$ asks to minimize $f_{\text{in}}(V) - \sum_{i=1}^{k} f_{\text{in}}(V_i) = \sum_{i=1}^{k}(\frac{f_{\text{in}}(V)}{k} - f_{\text{in}}(V_i))$. It is easy to see that both functions $g_1 = f - \frac{f(V)}{k}$ and $g_2 = \frac{f_{\text{in}}(V)}{k} - f_{\text{in}}$ are submodular, but may not be nonnegative or symmetric. Nevertheless, since both Theorem 2 and Lemma 1 do not require the function to be nonnegative or symmetric, we can still use GSA to find a k-partition and use Lemma 1 to estimate the performance. By easy calculations, we can enjoy the next performance guarantees.

Corollary 4 *The k-PPH with cost function $cost_1$ (resp., $cost_2$) can be approximated within factor $2 - \frac{2}{k}$ (resp., $\min(k, d^+_{\max})(1 - \frac{1}{k})$) in polynomial time, where $d^+_{\max}$ is the maximum degree of hyperedges with positive cost.* □

3 Greedy Splitting via Minimum 2, 3-Partitions

We have seen the GSA that increases the number of partitions one by one via minimum 2-partition computations. In this section we consider to increase the number of partitions two by two greedily. Let $k = 2m + 1 \geq 3$ be an odd number. (The case that k is an even number will be treated later.) We consider the next approximation algorithm for the k-PPSS. GSA2 (Table 2) contains m rounds and the i-th round constructs an $(2i + 1)$-partition $\mathcal{P}_{i+1}$ of V, where $\mathcal{P}_1 = \{V\}$, and the $(2i + 1)$-partition $\mathcal{P}_{i+1}$ is obtained by greedily "splitting" some member(s) in $\mathcal{P}_i$ at the minimum cost. There are two ways of such splitting. One is to split two members into four, which is considered by step 3. Another is to split

Table 2: Greedy splitting algorithm 2 (GSA2) for the k-PPSS with odd $k = 2m + 1$.

1 $\mathcal{P}_1 \leftarrow \{V\}$;
2 **for** $i = 1, \ldots, m$ **do**
3 $\quad (S_i^1, W_i^1, S_i^2, W_i^2) \leftarrow \text{argmin}\{\sum_{j=1}^{2}(f(S^j) + f(W^j - S^j) - f(W^j)) \mid$
$\qquad \emptyset \neq S^j \subset W^j,\ j = 1, 2, \text{ for distinct } W^1, W^2 \in \mathcal{P}_i\}$;
4 $\quad (T_i^1, T_i^2, W_i^3) \leftarrow \text{argmin}\{f(T^1) + f(T^2) + f(W - T^1 - T^2) - f(W) \mid$
$\qquad \{T^1, T^2, W - T^1 - T^2\} \text{ is a 3-partition of some } W \in \mathcal{P}_i\}$;
5 $\quad$ **if** $\sum_{j=1}^{2}(f(S_i^j) + f(W_i^j - S_i^j) - f(W_i^j)) < f(T_i^1) + f(T_i^2) + f(W_i^3 - T_i^1 - T_i^2) - f(W_i^3)$ **then**
6 $\qquad \mathcal{P}_{i+1} \leftarrow (\mathcal{P}_i - \{W_i^1, W_i^2\}) \cup \{S_i^1, W_i^1 - S_i^1, S_i^2, W_i^2 - S_i^2\}$;
7 $\quad$ **else**
8 $\qquad \mathcal{P}_{i+1} \leftarrow (\mathcal{P}_i - \{W_i^3\}) \cup \{T_i^1, T_i^2, W_i^3 - T_i^1 - T_i^2\}$;
9 $\quad$ **end** /* if */
10 **end** /* for */
11 Output $\mathcal{P}_{m+1}$.

one member into three, which is considered by step 4. We choose the one with the minimum cost to get $\mathcal{P}_{i+1}$ from $\mathcal{P}_i$ (step 5 – 8).

Clearly the output $\mathcal{P}_{m+1}$ is a k-partition of V. Let us consider the running time. In step 3 the objective is minimized by the least two minimum 2-partitions of members in $\mathcal{P}_i$. Thus by Theorem 2 step 3 can be done in $\sum_{W \in \mathcal{P}_i} O(|W|^3\theta) = O(|V|^3\theta)$ time. However, by now we do not know how to find a minimum 3-partition in submodular systems, which means that the time complexity of step 4 is still unknown in general. Therefore we suppose that the input (V, f) satisfies the next condition, which ensures that GSA2 runs in polynomial time.

Condition 1 *For any $W \subseteq V$, a 3-partition $\{T^1, T^2, W - T^1 - T^2\}$ of W that minimizes $f(T^1) + f(T^2) + f(W - T^1 - T^2)$ can be found in polynomial time.*

To analyze the performance of GSA2, we show a lemma analogous with Lemma 1.

Lemma 3 *For an $\ell \in \{0, 1, \ldots, m\}$, let $\mathcal{P}_{\ell+1}$ be the $(2\ell + 1)$-partition of V found by GSA2 in the ℓ-th round. Then for any ordered $(2\ell + 1)$-partition $\{V_1, V_2, \ldots, V_{2\ell+1}\}$ of V, it holds that*

$$f(\mathcal{P}_{\ell+1}) \leq \sum_{i=1}^{\ell}(f(V_{2i-1}) + f(V_{2i}) + f(V - V_{2i-1} - V_{2i})) - (\ell - 1)f(V). \tag{6}$$

Proof. We proceed by induction on ℓ. It is trivial when $\ell = 0$. Suppose that it holds for $\ell - 1$. Consider an ordered $(2\ell + 1)$-partition $\mathcal{P} = \{V_1, V_2, \ldots, V_{2\ell+1}\}$ of V. Since $\mathcal{P}_\ell$ is a $(2\ell - 1)$-partition of V, we see that at least one of the next two cases occurs for $\mathcal{P}$ and $\mathcal{P}_\ell$.

1. There is a $W^1 \in \mathcal{P}_\ell$ and three distinct $V_r, V_s, V_t \in \mathcal{P}$ $(r < s < t)$ such that $W^1 \cap V_r \neq \emptyset$, $W^1 \cap V_s \neq \emptyset$, and $W^1 \cap V_t \neq \emptyset$.

2. There are two distinct $W^1, W^2 \in \mathcal{P}_\ell$ and four distinct $V_a, V_b, V_p, V_q \in \mathcal{P}$ $(a < b,\ p < q,\ a < p)$ such that $W^1 \subseteq V_a \cup V_b$, $W^2 \subseteq V_p \cup V_q$, $W^1 \cap V_a \neq \emptyset \neq W^1 \cap V_b$ and $W^2 \cap V_p \neq \emptyset \neq W^2 \cap V_q$.

In case 1, we further consider the following two sub-cases.

1a. There is an $h \in \{1, \ldots, \ell\}$ such that $r = 2h - 1$ and $s = 2h$.

1b. Otherwise $r \in \{2h - 1, 2h\}$ and $s \in \{2h' - 1, 2h'\}$ for some $1 \leq h < h' \leq \ell$.

Similarly, we consider the next two sub-cases in case 2.

2a. There is an $h \in \{1, \ldots, \ell\}$ such that $a = 2h - 1$ and $p = 2h$.

2b. Otherwise $a \in \{2h - 1, 2h\}$ and $p \in \{2h' - 1, 2h'\}$ for some $1 \leq h < h' \leq \ell$.

We will show that in each sub-case of 1a, 1b, 2a, 2b, there is a "nice splitting" which is a candidate for step 3 or 4 of GSA2. (Recall that the cost of any "nice splitting" is an upper bound on $f(\mathcal{P}_{\ell+1}) - f(\mathcal{P}_\ell)$.) We show that we can construct an ordered $(2\ell-1)$-partition $\mathcal{P}' = \{V'_1, \ldots, V'_{2\ell-1}\}$ of V from $\mathcal{P}$ such that $\sum_{i=1}^{\ell-1} \left(f(V'_{2i-1}) + f(V'_{2i}) + f(V - V'_{2i-1} - V'_{2i})\right) - (\ell-2)f(V)$ plus the cost of the "nice splitting" is at most the right hand of (6). This will prove the lemma by the induction hypothesis on $\mathcal{P}'$.

In what follows, we only consider sub-case 2a due to space limitation (the other cases can be shown analogously). Let $\mathcal{P}'$ be the ordered $(2\ell-1)$-partition $\{V_1, \ldots, V_{2h-2}, V_{2h+1}, \ldots, V_{2\ell}, V_{2h-1} \cup V_{2h} \cup V_{2\ell+1}\}$ of V, which has the same order as $\mathcal{P}$ except for that V_{2h-1} and V_{2h} are merged with the last member $V_{2\ell+1}$ (notice $2h-1 < 2h < 2\ell+1$). By the induction hypothesis on $\ell-1$, (6) holds for $\mathcal{P}_\ell$ and $\mathcal{P}'$, i.e.,

$$f(\mathcal{P}_\ell) \;\le\; \sum_{1 \le i \le \ell,\; i \ne h} \left(f(V_{2i-1}) + f(V_{2i}) + f(V - V_{2i-1} - V_{2i})\right) - (\ell-2)f(V).$$

Thus, it suffices to show

$$f(\mathcal{P}_{\ell+1}) - f(\mathcal{P}_\ell) \;\le\; f(V_{2h-1}) + f(V_{2h}) + f(V - V_{2h-1} - V_{2h}) - f(V). \tag{7}$$

For this, we choose $(W^1 \cap V_{2h-1}, W^1, W^2 \cap V_{2h}, W^2)$ as the "nice splitting", i.e., split W^1 and W^2 into $W^1 \cap V_{2h-1}$, $W^1 - V_{2h-1}$ and $W^2 \cap V_{2h}$, $W^2 - V_{2h}$ respectively. Clearly it is a candidate for $(S^1_\ell, W^1_\ell, S^2_\ell, W^2_\ell)$ in step 3 of GSA2 (see Table 2). Therefore,

$$f(\mathcal{P}_{\ell+1}) - f(\mathcal{P}_\ell) \le f(W^1 \cap V_{2h-1}) + f(W^1 - V_{2h-1}) - f(W^1) + f(W^2 \cap V_{2h}) + f(W^2 - V_{2h}) - f(W^2). \tag{8}$$

By the submodularity, the right hand of (8) is at most

$$\begin{aligned}
& f(W^1 \cap V_{2h-1}) - f(W^1) + f(W^2 \cap V_{2h}) + f(W^2 \cup V_{2h}) - f(W^2) - f(V) \\
& \quad + [f(W^1 - V_{2h-1}) + f(V - V_{2h})] \\
& \le \; [f(W^1 \cap V_{2h-1}) + f(W^1 \cup V_{2h-1}) - f(W^1)] \\
& \quad + [f(W^2 \cap V_{2h}) + f(W^2 \cup V_{2h}) - f(W^2)] + f(V - V_{2h-1} - V_{2h}) - f(V) \\
& \le \; f(V_{2h-1}) + f(V_{2h}) + f(V - V_{2h-1} - V_{2h}) - f(V),
\end{aligned}$$

proving (7). □

For an even $k = 2m \ge 2$, we start with a minimum 2-partition of V before increasing the number of partitions two by two greedily. It is described in Table 3, where the same code as in Table 2 are abbreviated.

Table 3: Greedy splitting algorithm 2 (GSA2) for the k-PPSS with even $k = 2m$.

1	$\mathcal{P}_1 \leftarrow$ a minimum 2-partition of V;
2	**for** $i = 1, \ldots, m-1$ **do**
3 – 10	(The same as step 3 – 10 in Table 2);
11	Output $\mathcal{P}_m$.

Clearly the output $\mathcal{P}_m$ is a k-partition of V. In order to be a polynomial time algorithm, it is again assumed that Condition 1 is satisfied. We give a lemma on the performance, where the proof can be done in a similar way as Lemma 3 and is omitted.

Lemma 4 *For an $\ell \in \{1, 2, \ldots, m\}$, let $\mathcal{P}_\ell$ be the 2ℓ-partition of V found by GSA2 in the ℓ-th round. Then for any ordered 2ℓ-partition $\{V_1, V_2, \ldots, V_{2\ell}\}$ of V, it holds that*

$$f(\mathcal{P}_\ell) \;\le\; f(V_1) + f(V - V_1) + \sum_{i=1}^{\ell-1} \left(f(V_{2i}) + f(V_{2i+1}) + f(V - V_{2i} - V_{2i+1})\right) - (\ell-1)f(V). \tag{9}$$

□

We note that, not surprisingly, GSA2 does no worse than GSA in any cases. This can be seen by comparing the right hand of (2) and (6) or (9). Notice that $f(V - A - B) + f(V) \le f(V - A) + f(V - B)$ for any disjoint subsets A and B of V. In fact, using Lemma 3 and Lemma 4, we have the next result.

Theorem 6 *The performance guarantee of GSA2 is $\frac{k-1}{2}$ for the k-PPSS and $2-\frac{2}{k}$ for the k-PP3S. There are examples that indicate these bounds are tight.* □

We know that GSA2 can do better for the k-PPG [19]. A question is, what can it guarantee to problem classes lying between the k-PPG and the k-PP3S e.g., the k-PPH. In the following, we show that GSA2 achieves a guarantee better than $2-\frac{2}{k}$ for the k-PPH, extending the result for the k-PPG by [19].

Theorem 7 *The k-PPH can be approximated in polynomial time within factor $2-\frac{3}{k}$ for any odd $k \geq 3$ and factor $2-\frac{3}{k}+\frac{1}{k^2-k}$ for any even $k \geq 2$.*

Proof. Let V and f be respectively the vertex set and the cut function of a hypergraph H. It is easy to see that Condition 1 is satisfied by considering the reduced hypergraph of H for any vertex subset $W \subseteq V$, where for each hyperedge e, the endpoints of e that are not in W are removed and e is present if it has at least two endpoints in W. Thus GSA2 (Table 2, 3) is a polynomial time approximation algorithm for the k-PPH. We next show the claimed performance guarantee. Let $\mathcal{P}^* = \{V_1^*, V_2^*, \ldots, V_k^*\}$ be a minimum k-partition of V. Let π denote a numbering of $\{1, \ldots, k\}$, and let $\pi(i)$ be the number of i.

First consider an odd number $k = 2m+1 \geq 3$. By applying Lemma 3 to $\mathcal{P}^*$, we see that GSA2 finds a k-partition of V with cost at most $f_\pi = \sum_{i=1}^m (f(V^*_{\pi(2i-1)}) + f(V^*_{\pi(2i)}) + f(V - V^*_{\pi(2i-1)} - V^*_{\pi(2i)}))$ for any numbering π. We want to show that there is a numbering π^* such that f_{π^*} is no more than $2-\frac{3}{k}$ times the optimum $f(\mathcal{P}^*) = \sum_{i=1}^k f(V_i^*)$. This can be done by considering all numberings and showing that the average value of f_π is at most $(2-\frac{3}{k})f(\mathcal{P}^*)$.

Let us rewrite f_π as $2f(P^*) - \Delta_\pi$ where $\Delta_\pi = 2f(V^*_{\pi(k)}) + \sum_{i=1}^m (f(V^*_{\pi(2i-1)}) + f(V^*_{\pi(2i)}) - f(V - V^*_{\pi(2i-1)} - V^*_{\pi(2i)}))$. Thus we only need to show that the average value of Δ_π is at least $\frac{3}{k} f(\mathcal{P}^*)$. For each hyperedge e, we consider the average number that e is counted in Δ_π. For simplicity, let us contract each $V_i^* \in \mathcal{P}^*$ to a single node v_i (this may decrease the degree of e). Let $H|_{\mathcal{P}^*}$ denote the contracted hypergraph. To avoid confusing we use the word "node" in $H|_{\mathcal{P}^*}$ to denote the contracted vertex subsets. We assume without loss of generality that $H_{\mathcal{P}^*}$ is simple and complete. Otherwise we can meet this by merging the hyperedges with the same endpoints and adding zero cost hyperedges. Suppose that after contraction e has degree $d \geq 2$ (otherwise e is not counted in Δ_π).

Recall that $f(S)$ is the sum of costs of hyperedges that has at least one but not all endpoints in S for $S \subseteq V$. Thus due to the $2f(V^*_{\pi(k)})$ term in Δ_π, e is counted twice if one endpoint of e is numbered k. Since $H|_{\mathcal{P}^*}$ has k nodes and e has d endpoints, we see that the average number (expected value) that e is counted due to the $2f(V^*_{\pi(k)})$ term is $\frac{2d}{k}$. On the other hand, due to the other term $\sum_{i=1}^m (f(V^*_{\pi(2i-1)}) + f(V^*_{\pi(2i)}) - f(V - V^*_{\pi(2i-1)} - V^*_{\pi(2i)}))$ in Δ_π, e is counted twice if $d=2$ and the two endpoints of e are numbered $\pi(2i-1)$ and $\pi(2i)$ for some $i \in \{1, 2, \ldots, m\}$, otherwise e is counted p times if $d \geq 3$ and the endpoints of e contains p pairs of nodes that are numbered $\pi(2i_p - 1)$ and $\pi(2i_p)$ for some distinct $i_p \in \{1, 2, \ldots, m\}$. Notice that for each fixed pair of indices $2i-1$ and $2i$, the average number (probability) that both nodes v_{2i-1} and v_{2i} become endpoints of e is $\binom{k-2}{d-2}/\binom{k}{d} = \frac{d(d-1)}{k(k-1)}$. Thus the average number that e is counted due to the $\sum_{i=1}^m (f(V^*_{\pi(2i-1)}) + f(V^*_{\pi(2i)}) - f(V - V^*_{\pi(2i-1)} - V^*_{\pi(2i)}))$ term is $2 \cdot m \cdot \frac{2\cdot(2-1)}{k(k-1)} = \frac{2}{k}$ if $d=2$, or $m \cdot \frac{d(d-1)}{k(k-1)} = \frac{d(d-1)}{2k}$ if $d \geq 3$. Since e is counted d times in the optimum $f(\mathcal{P}^*)$, we see that the contribution of e to the average value of Δ_π is $\frac{1}{2}(\frac{4}{k} + \frac{2}{k}) = \frac{3}{k}$ $(d=2)$ or $\frac{1}{d}(\frac{2d}{k} + \frac{d(d-1)}{2k}) \geq \frac{3}{k}$ $(d \geq 3)$ times the contribution to $f(\mathcal{P}^*)$. Thus we see that the average value of Δ_π is at least $\frac{3}{k}$ times $f(\mathcal{P}^*)$, which finishes the proof of the theorem for an odd k.

Similarly, we can prove the theorem for an even k. We note that the bounds are tight, see [19]. □

4 Target Split Problem in Submodular Systems

Given a *target set* $T \subseteq V$ ($|T| \geq k$) as an additional input, the *target split problem in submodular systems* (TSPSS) is to find a minimum k-partition $\{V_1, V_2, \ldots, V_k\}$ such that each V_i contains at least one target. By considering only "valid" k-partitions (i.e., a target split of T), we extend our algorithms to TSPSS.

Let us first consider algorithm GSA. In step 3 of GSA, we need to compute a valid 2-partition for some W in the current solution $\mathcal{P}_i$ at the minimum cost. This can be done if we can compute a minimum valid 2-partition for each $W \in \mathcal{P}_i$. We do this in the next way.

We do nothing with $W \in \mathcal{P}_i$ such that $|T \cap W| \leq 1$. For each $W \in \mathcal{P}_i$ with $|T \cap W| \geq 2$, we choose a target $s \in T \cap W$, compute a minimum 2-partition of W that separates s and t for each target $t \in T \cap W - \{s\}$, and choose the one with the minimum cost. We see that a minimum 2-partition of W that separates specified vertices s and t can be found in polynomial time.

Lemma 5 *Given a submodular system (V, f) and a $W \subseteq V$, for any $s, t \in W$ $(s \neq t)$, a subset S^* of W such that $s \in S^*$, $t \notin S^*$ and $g(S^*) + g(W - S^*)$ is minimum can be found in polynomial time.*

Proof. Consider a submodular system $(W - \{s,t\}, g)$ where $g(S) = f(S \cup \{s\}) + f(W - S - \{s\})$ for all $S \subseteq W - \{s,t\}$. Clearly we need only to find a subset S' of $W - \{s,t\}$ that $g(S')$ is minimum by letting $S^* = S' \cup \{s\}$. Since g is submodular, it can be minimized in polynomial time [5, 17]. □

Hence we have seen that GSA can be extended to the TSPSS and runs in polynomial time. Furthermore, the performance can still be shown in a straightforward manner as Lemma 1 and Theorem 3. We summarize this as the next theorem.

Theorem 8 *Given a nonnegative submodular system (V, f) with a target set $T \subseteq V$, the TSPSS can be approximated within factor $(1+\alpha)(1-\frac{1}{k})$ in polynomial time, where α is any number 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 that is a target split of T, where we can let $\alpha = k-1$ in general, and let $\alpha = 1$ for symmetric f.* □

Let us consider GSA2. Since the multiterminal cut problem is NP-hard even for $k = 3$, we cannot expect a polynomial time algorithm to compute a minimum 3-partition that is a target split in general (unless P=NP). Nevertheless, we note that Lemma 3, 4 can be extended in a straightforward manner.

5 Conclusion and Remark

In this paper, we have presented a simple and unified approach for developing and analyzing approximation algorithms for some multiway partition related minimization problems. The main idea is a greedy splitting approach to unified problems k-PPSS (k-partition problem in submodular systems) and TSPSS (target split problem in submodular systems). Several important and interesting results are shown in this paper. We note that it is still open whether the k-PPSS can be solved in polynomial time for any $k \geq 3$ (the 2-PPSS is shown to be solvable in polynomial time). Finally, we remark that it seems not so easy as in this paper to show the performance guarantee of greedy algorithms that increase the number of partitions three (or more) by three (or more). This is because analogous properties that we have shown in Lemma 1, 3 and 4 no longer hold even for k-PPG [19].

References

[1] G. Călinescu, H. Karloff and Y. Rabani. An Improved Approximation Algorithm for Multiway Cut. In Proc. STOC 1998, pp. 48–52, 1998.

[2] S. Chopra and J. H. Owen. A Note on Formulations for the A-Partition Problem on Hypergraphs. Discrete Applied Mathematics 90, pp. 115–133, 1999.

[3] E. Dalhaus, D. S. Johnson, C. H. Papadimitriou, P. Seymour and M. Yannakakis. The Complexity of the Multiterminal Cuts. SIAM J. Comput. Vol. 23, No. 4, pp. 864–894, 1994.

[4] O. Goldschmidt and D. S. Hochbaum. A Polynomial Algorithm for the k-cut Problem for fixed k. Math. Oper. Res. 19, No. 1, pp. 24–37, 1994.

[5] S. Iwata, L. L.Fleischer and S. Fujishige. A Combinatorial, Strongly Polynomial-Time Algorithm for Minimizing Submodular Functions. In Proc. STOC 2000, pp. 97–106, 2000.

[6] S. Kapoor. On Minimum 3-cuts and Approximating k-cuts Using Cut Trees. IPCO 1996, LNCS 1084, pp. 132–146, 1996.

[7] D. R. Karger, P. Klein, C. Stein, M. Thorup and N. E. Young. Rounding Algorithms for a Geometric Embedding of Minimum Multiway Cut. In Proc. STOC 1999, pp. 668–678, 1999.

[8] D. R. Karger and C. Stein. A New Approach to the Minimum Cut Problems. J. of ACM 43, No. 4, pp. 601–640, 1996.

[9] T. Lengauer. Combinatorial Algorithms for Integrated Circuit Layout. Wiley, New York, 1990.

[10] D. W. Matula, A Linear Time $2+\epsilon$ Approximation Algorithm for Edge Connectivity. In Proc. SODA 1993, pp. 500–504, 1993.

[11] N. Maeda, H. Nagamochi, T. Ibaraki. Approximate Algorithms for Multiway Objective Point Split Problems of Graphs (Japanese). Computing Devices and Algorithms (Japanese) (Kyoto, 1993). Surikaisekikenkyusho Kokyuroku No. 833, pp. 98–109, 1993.

[12] H. Nagamochi and T. Ibaraki. A Fast Algorithm for Computing Minimum 3-way and 4-way Cuts. Mathematical Programming 88, No. 3, pp. 507–520, 2000.

[13] H. Nagamochi, S. Katayama and T. Ibaraki. A Faster Algorithm for Computing Minimum 5-way and 6-way Cuts in Graphs. J. Combinatorial Optimization 4, pp. 35–78, 2000.

[14] H. Narayanan, S. Roy and S. Patkar. Approximation Algorithms for Min-k-Overlap Problems Using the Principal Lattice of Partitions Approach. J. Algorithms 21, pp. 306–330, 1996.

[15] M. Queyranne. Minimizing Symmetric Submodular Functions. Mathematical Programming B 82, pp. 3–12, 1998.

[16] M. Queyranne. On Optimum Size-Constrained Set Partitions. AUSSOIS 1999, France, March 8–12, 1999.

[17] A. Schrijver. A Combinatorial Algorithm Minimizing Submodular Functions in Strongly Polynomial Time. J. Combin. Theory B 80, No. 2, pp. 346–355, 2000.

[18] H. Saran and V. V. Vazirani. Finding k-cuts within Twice the Optimal. SIAM J. Comput. 24, No. 1, pp. 101–108, 1995.

[19] L. Zhao, H. Nagamochi and T. Ibaraki. Approximating the Minimum k-way Cut in a Graph via Minimum 3-way Cuts. ISAAC 1999, LNCS 1741, pp. 373–382, 1999. Full paper will appear in J. Combinatorial Optimization.