アルゴリズム 66−5
(1999. 1. 27)

最小 3-カット，4-カットを計算するアルゴリズム

永持 仁　　茨木 俊秀
京都大学 情報学研究科 数理工学専攻
京都市左京区吉田本町 606-8501
{naga,ibaraki}@kuamp.kyoto-u.ac.jp

あらまし　$k=3,4$ に対し，重み付きグラフの最小 k-カットを求める $O(n^{k-2}(nF(n,m)+C_2(n,m)+n^2))=O(mn^k\log(n^2/m))$ 時間アルゴリズムを与える．ここで、n, m はグラフの点数，辺数であり，$F(n,m)$ と $C_2(n,m)$ はそれぞれ n 点、m 辺グラフ上での最大流，最小 2-カットを求める時間である．このアルゴリズムを拡張させて，対称な劣モジュラシステムにおける最小 3-カットを効率良く計算することもできる．

A Fast Algorithm for Computing Minimum 3-Way and 4-Way Cuts

Hiroshi Nagamochi and Toshihide Ibaraki
Dept. of Applied Mathematics and Physics
Graduate School of Informatics
Kyoto University, Kyoto, Japan 606-8501
{naga,ibaraki}@kuamp.kyoto-u.ac.jp

Abstract For an edge-weighted graph G with n vertices and m edges, we present a new deterministic algorithm for computing a minimum k-way cut for $k=3,4$. The algorithm runs in $O(n^{k-2}(nF(n,m)+C_2(n,m)+n^2))=O(mn^k\log(n^2/m))$ time for $k=3,4$, where $F(n,m)$ and $C_2(n,m)$ denote respectively the time bounds required to solve the maximum flow problem and the minimum 2-way cut problem in G. The bound for $k=3$ matches the current best deterministic bound $\tilde{O}(mn^3)$ for weighted graphs, but improves the bound $\tilde{O}(mn^3)$ to $O(n(nF(n,m)+C_2(n,m)+n^2))=O(\min\{mn^{8/3},m^{3/2}n^2\})$ for unweighted graphs. The bound $\tilde{O}(mn^4)$ for $k=4$ improves upon both the previous best deterministic bound $\tilde{O}(mn^5)$ and randomized bound $\tilde{O}(n^6)$ (for $m=o(n^2)$). The algorithm is then generalized to the problem of finding a minimum 3-way cut in a symmetric submodular system.

1 Introduction

Let G stand for an undirected graph with its edges being weighted by non-negative real numbers, and let n and m denote the numbers of vertices and edges, respectively. For an integer $k\geq 2$, a *k-way cut* is a partition $\{V_1,V_2,\ldots,V_k\}$ of V consisting of k non-empty subsets. The problem of partitioning V into k non-empty subsets so as to minimize the weight sum of the edges between different subsets is called *the minimum k-way cut problem.* The problem has several important applications such as VLSI design [14], task allocation in distributed computing systems and network reliability. The 2-way cut problem (i.e., the problem of computing the edge-connectivity) can be solved in $\tilde{O}(nm)$ time [6, 15]. For $k=3$, Hochbaum and Shmoys [7] proved that the problem in an unweighted planar graph can be solved in $O(n^2)$ time. However, the complexity status of the problem for general $k\geq 3$ in an arbitrary graph G has been open for several years. Goldschmidt and Hochbaum proved that the problem is NP-hard if k is an input parameter [5]. In the same article [5], they presented an $O(n^{k^2/2-3k/2+4}F(n,m))$ time algorithm for solving the minimum k-way cut problem, where $F(n,m)$ denotes the time required to find a minimum (s,t)-cut (i.e., a minimum 2-way cut that separates two specified vertices s and t) in an edge-weighted graph with n vertices and m edges, which can be obtained by applying a maximum flow algorithm. This running time is polynomial for any fixed k. Afterwards, Karger and Stein [11] proposed a randomized algorithm that solves the minimum k-way cut problem with high probability in $O(n^{2(k-1)}(\log n)^3)$ time.

For $k=3$, Kapoor [10] and Kamidoi *et al.* [8] showed that the problem can be solved in $O(n^3F(n,m))$ time, which was then improved to $\tilde{O}(mn^3)$ by Burlet and Goldschmidt [1] and

Kamidoi *et al.* [9]. For $k = 4$, Kamidoi *et al.* [8] gave an $O(n^4 F(n,m)) = \tilde{O}(mn^5)$ time algorithm, and they recently proposed an $O(n^{2(k-2)}(F(n,m) + C_2(n,m)))$ time algorithm for the minimum k-way cut problem [9], where $C_2(n,m)$ denotes the time required to find a minimum 2-way cut in an edge-weighted graph with n vertices and m edges.

Let us call a non-empty and proper subset X of V a *cut*. Clearly, if we can identify the first cut V_1 in a minimum k-way cut $\{V_1, \ldots, V_k\}$, then the rest of cuts $V_2, \ldots, V_k$ can be obtained by solving the minimum $(k-1)$-way cut problem in the graph induced by $V - V_1$. The articles [1, 9] succeeded to characterize a set of $O(n^2)$ number of cuts which contains at least one such cut V_1. Thus, by solving $O(n^2)$ minimum $(k-1)$-way cut problems, a minimum k-way cut can be computed. For example, Burlet and Goldschmidt [1] showed that, given a minimum 2-way cut $\{X, V - X\}$ in G, such V_1 is a cut whose weight is less than 4/3 of the weight of a minimum 2-way cut in G or in the induced subgraphs $G[X]$ and $G[V - X]$. Since it is known [17] that there are $O(n^2)$ cuts with weight less than 4/3 of the weight of a minimum 2-way cut, and all those cuts can be enumerated in $\tilde{O}(mn^3)$ time, their approach yields an $\tilde{O}(mn^3)$ time minimum 3-way cut algorithm.

In this paper, we consider the minimum k-way cut problem for $k = 3, 4$, and give a new characterization of a set of $O(n)$ number of such candidate cuts for the first cut V_1, on the basis of the submodularity of cut functions. We also show that those $O(n)$ cuts can be obtained in $O(n^2 F(n,m))$ time by using Vazirani and Yannakakis's algorithm for enumerating small cuts [19]. Therefore, we can find a minimum 3-way cut in $O(n^2 F(n,m) + nC_2(n,m)) = O(mn^3 \log(n^2/m))$ time and a minimum 4-way cut in $O(n^3 F(n,m) + n^2 C_2(n,m)) = O(mn^4 \log(n^2/m))$ time. The bound for $k = 3$ matches the current best deterministic bound $\tilde{O}(mn^3)$ for weighted graphs, but improves the bound $\tilde{O}(mn^3)$ to $O(\min\{mn^{8/3}, m^{3/2}n^2\})$ for unweighted graphs (since $F(n,m) = O(\min\{mn^{2/3}, m^{3/2}\})$ is known for unweighted graphs [2]). The bound $\tilde{O}(mn^4)$ for $k = 4$ improves upon both the previous best deterministic bound $\tilde{O}(mn^5)$ and randomized bound $\tilde{O}(n^6)$ (for $m = o(n^2)$). In the case of an edge-weighted planar graph G, we also shows that the algorithm can be implemented to run in $O(n^3)$ time for $k = 3$ and in $O(n^4)$ time for $k = 4$, respectively. The algorithm is then generalized to the problem of finding a minimum 3-way cut in a symmetric submodular system.

2 Preliminaries

2.1 Notations and definitions

A singleton set $\{x\}$ may be simply written as x, and " $\subset$ " implies proper inclusion while " $\subseteq$ " means " $\subset$ " or " $=$ ". For a finite set V, a *cut* is defined as a non-empty and proper subset X of V. For two disjoint subsets $S, T \subset V$, we say that a cut X *separates* S and T if $S \subseteq X \subseteq V - T$ or $T \subseteq X \subseteq V - S$ holds. A subset $X \subseteq V$ *intersects* another subset $Y \subseteq V$ if $X - Y \neq \emptyset$, $Y - X \neq \emptyset$ and $X \cap Y \neq \emptyset$ hold, and X *crosses* Y if, in addition, $V - (X \cup Y) \neq \emptyset$ holds. A family $\mathcal{X}$ of subsets of V is called *non-intersecting* (resp., *non-crossing*) if no two subsets $X, Y \in \mathcal{X}$ intersect (resp., cross) each other It is possible that a non-crossing family $\mathcal{X}$ contains a pair of intersecting subsets.

The following observation plays a key role in analyzing the time bound of our algorithm. We easily observe that $O(n)$ cuts belongs to a non-crossing set of cuts.

Lemma 1 *Let V be a non-empty finite set, and $\mathcal{X}$ be a family of cuts in V such that, for each subset $X \in \mathcal{X}$, its complement $V - X$ does not belong to $\mathcal{X}$. Then $|\mathcal{X}| \leq 2|V| - 3$.* □

A *set function* f on a ground set V is a function $f : 2^V \mapsto \Re$, where $\Re$ is the set of real numbers. A set function f is called *submodular* if

$$f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y) \tag{1}$$

holds for every pair of subsets $X, Y \subseteq V$. An f is called *symmetric* if

$$f(X) = f(V - X) \text{ for all subsets } X \subseteq V. \tag{2}$$

For a symmetric and submodular set function f, it holds

$$f(X) + f(Y) \geq f(X - Y) + f(Y - X) \tag{3}$$

for every pair of subsets $X, Y \subseteq V$. A pair (V, f) of a finite set V and a set function f on V is called a *system*. It is called a *symmetric submodular* system if f is symmetric and submodular.

Given a system (V, f), we define the minimum k-way cut problem as follows. A *k-way cut* is a partition $\pi = \{V_1, V_2, \ldots, V_k\}$ of V consisting of k non-empty subsets. The weight $\omega_f(\pi)$ of a k-way cut π is defined by

$$\omega_f(\pi) = \frac{1}{2}(f(V_1) + f(V_2) + \cdots + f(V_k)). \tag{4}$$

A k-way cut is called *minimum* if it has the minimum weight among all k-way cuts in (V, f).

Let $G = (V, E)$ be an undirected graph with a set V of vertices and a set E of edges weighted by non-negative reals. For a non-empty subset $X \subseteq V$, let $G[X]$ denote the graph induced from G by X. For a subset $X \subset V$, its *cut value*, denoted by $c(X)$, is defined to be the sum of weights of edges between X and $V - X$, where $c(\emptyset)$ and $c(V)$ are defined to be 0. This set function c on V is called the *cut function* of G. The cut function c is symmetric and submodular, as easily verified. For a k-way cut $\pi = \{V_1, V_2, \ldots, V_k\}$ of V, its weight in G is defined by $\omega_c(\pi) = \frac{1}{2}(c(V_1) + c(V_2) + \cdots + c(V_k))$, which means the weight sum of the edges between different cuts in π.

2.2 Enumerating all 2-way cuts

In [19], Vazirani and Yannakakis presented an algorithm that finds all the 2-way cuts in G in the order of non-decreasing weights. The algorithm finds the next 2-way cut by solving at most $2n - 3$ maximum flow problems. We describe this result in a slightly more general way in terms of set functions. This algorithm will be used in Section 3 to obtain minimum 3-way and 4-way cuts.

Theorem 1 [19] *For a system (V, f) with $n = |V|$, 2-way cuts in (V, f) can be enumerated in the order of non-decreasing weights with $O(nF_f)$ time delay between two consecutive outputs, where F_f is the time required to find a minimum 2-way cut that separates specified two disjoint subsets $S, T \subset V$ in (V, f).* □

It is known that a non-empty and proper subset X that minimizes $g(X)$ in a submodular system (V, g) can be found in polynomial time by using the ellipsoid method [4]. In particular, the problem of finding a minimum 2-way cut which separates specified two disjoint subsets $S, T \subset V$ in (V, f) can be solved in polynomial time, since the problem is reduced to finding a minimum 2-way cut in the system (V', g) such that $V' = V - (S \cup T)$ and a submodular function $g : V' \to \Re$ defined by $g(X) = f(X \cup S) - f(S)$.

For the cut function c in an edge-weighted graph G, it is well known that a minimum 2-way cut that separates specified two disjoint subsets $S, T \subset V$ can be found by solving a single maximum flow problem. Therefore $F_f = O(mn \log(n^2/m))$ holds if we use the maximum flow algorithm of [3], where n and m are the numbers of vertices and edges in G, respectively. In the planar graph case, we can enumerates cuts more efficiently by making use of dual graphs.

Call a simple path between s and t an *s, t-path*, and let $S(n, m)$ denote the time to compute a shortest s, t-path in an edge-weighted graph G with n vertices and m edges.

Lemma 2 *For an edge-weighted graph $G = (V, E)$ with $n = |V|$ and $m = |E|$, cycles in G can be enumerated in the order of non-decreasing lengths, with $O(S(n, m))$ time delay between two consecutive outputs, where the first cycle can be found in $O(mS(n, m))$ time.* □

Corollary 1 *For an edge-weighted planar graph $G = (V, E)$ with $n = |V|$, 2-way cuts in G can be enumerated in the order of non-decreasing weights with $O(n)$ time delay between two consecutive outputs, where the first 2-way cut can be found in $O(n^2)$ time.* □

3 Minimum 3-Way and 4-Way Cuts

Let $\pi = \{V_1, \ldots, V_k\}$ be a minimum k-way cut in a system (V, f). Suppose that the first cut V_1 in π is known. Then we need to compute a minimum k-way cut $\{V_1, \ldots, V_k\}$ in (V, f) under the restriction that V_1 is fixed. In the case of a graph, the problem can be reduced to finding a minimum $(k-1)$-way cut in the induced subgraph $G[V - V_1]$. For $k = 3, 4$, we now show that such a cut V_1 can be obtained from among $O(n)$ number of cuts.

Theorem 2 *For a symmetric submodular system (V, f) with $n = |V| \geq 4$, let*

$$\{X_1, V - X_1\}, \{X_2, V - X_2\}, \ldots, \{X_r, V - X_r\}$$

be the first r smallest 2-way cuts in the order of non-decreasing weights, where r is the first integer r such that X_r crosses some X_q $(1 \leq q < r)$. Let us denote $Y_1 = X_q - X_r$, $Y_2 = X_q \cap X_r$, $Y_3 = X_r - X_q$, and $Y_4 = V - (X_q \cup X_r)$. Then:

(i) *There is a minimum 3-way cut $\{V_1, V_2, V_3\}$ of (V, f) such that $V_1 = X_i$ or $V_1 = V - X_i$ for some $i \in \{1, 2, \ldots, r-1\}$, or $\{V_1, V_2, V_3\} = \{Y_j, Y_{j+1}, V - (Y_j \cup Y_{j+1})\}$ for some $j \in \{1, 2, 3, 4\}$ (where $Y_{j+1} = Y_1$ for $j = 4$).*

(ii) *There is a minimum 4-way cut $\{V_1, \ldots, V_4\}$ of (V, f) such that $V_1 = X_i$ or $V_1 = V - X_i$ for some $i \in \{1, 2, \ldots, r-1\}$, or $\{V_1, \ldots, V_4\} = \{Y_1, Y_2, Y_3, Y_4\}$.*

__Proof__. We have $f(X_1) \leq f(X_2) \leq \cdots \leq f(X_r)$ by assumption.

(i) Let a minimum 3-way cut in G be denoted by $\{V_1, V_2, V_3\}$ with $f(V_1) \leq \min\{f(V_2), f(V_3)\}$. If $f(V_1) < f(X_r)$, this implies that $V_1 = X_i$ or $V_1 = V - X_i$ for some $i < r$. Therefore assume $f(V_1) \geq f(X_r)$. Thus, $\omega_f(\{V_1, V_2, V_3\}) \geq \frac{3}{2} f(X_r)$ holds. Now the cuts X_q and X_r cross each other, and $f(X_q) \leq f(X_r)$ holds by $q < r$. By (3), we have $f(X_q) + f(X_r) \geq f(X_q - X_r) + f(X_r - X_q)$. Hence at least one of $f(X_q - X_r)$ and $f(X_r - X_q)$ is equal to or less than $\max\{f(X_q), f(X_r)\} (= f(X_r))$. Thus, $f(Y_i) \leq f(X_r)$ holds for $i = 1$ or $i = 3$. Assume that $f(X_h - X_j) \leq f(X_r)$ holds for $\{h, j\} = \{q, r\}$. Similarly, we obtain $\min\{f(X_q \cap X_r), f(X_q \cup X_r)\} \leq f(X_r)$ by (1). Thus, $f(Y_i) \leq f(X_r)$ holds for $i = 2$ or $i = 4$ (note that $f(X_q \cup X_r) = f(V - (X_q \cup X_r))$). Therefore, there is an index $j \in \{1, 2, 3, 4\}$ such that $f(Y_j) \leq f(X_r)$ and $f(Y_{j+1}) \leq f(X_r)$ (where Y_{j+1} implies Y_1 for $j = 4$). Clearly, $f(V - (Y_j \cup Y_{j+1}))$ is equal to $f(X_q)$ or $f(X_r)$. Therefore, for the 3-way cut $\{Y_j, Y_{j+1}, V - (Y_j \cup Y_{j+1})\}$, we have $\omega_f(\{Y_j, Y_{j+1}, V - (Y_j \cup Y_{j+1})\}) \leq \frac{3}{2} f(X_r) \leq \omega_f(\{V_1, V_2, V_3\})$. This implies that $\{Y_j, Y_{j+1}, V - (Y_j \cup Y_{j+1})\}$ is also a minimum 3-way cut.

(ii) Let a minimum 4-way cut in G be denoted by $\{V_1, V_2, V_3, V_4\}$ with $f(V_1) \leq \min\{f(V_2), f(V_3), f(V_4)\}$. Assume that $f(V_1) \geq f(X_r)$ holds, since otherwise (i.e., $f(V_1) < f(X_r)$), we are done. Thus, $2f(X_r) \leq 2f(V_1) \leq \omega_f(\{V_1, V_2, V_3, V_4\})$. From inequalities (1) - (3), we then obtain

$$2f(X_r) \geq f(X_q) + f(X_r) \geq \frac{1}{2}[f(X_q - X_r) + f(X_r - X_q) + f(X_q \cap X_r) + f(V - (X_q \cup X_r))].$$

Therefore, we have $\frac{1}{2}[f(X_q - X_r) + f(X_r - X_q) + f(X_q \cap X_r) + f(V - (X_q \cup X_r))] \leq f(X_q) + f(X_r) \leq 2f(X_r) \leq \omega_f(\{V_1, V_2, V_3, V_4\})$, indicating that $\{X_q - X_r, X_r - X_q, X_q \cap X_r, V - (X_q \cup X_r)\}$ is also a minimum 4-way cut in (V, f). □

Now we are ready to describe a new algorithm for computing minimum 3-way and 4-way cuts in an edge-weighted graph G. In the algorithm, minimum 2-way cuts will be stored in $\mathcal{X}$ in the non-decreasing order, until $\mathcal{X}$ becomes crossing, and an $O(n)$ number of k-way cuts will be stored in $\mathcal{C}$, from which a k-way cut of the minimum weight is chosen.

<u>MULTIWAY</u>

Input: An edge-weighted graph $G = (V, E)$ with $|V| \geq 4$ and an integer $k \in \{3, 4\}$.

Output: A minimum k-way cut π.

```
    𝒳 := 𝒞 := ∅; i := 1;
    while 𝒳 is non-crossing do
      Find the i-th minimum 2-way cut {X_i, V − X_i} in G;
      𝒳 := 𝒳 ∪ {X_i};
```

5 if $|V - X_i| \geq k-1$ **then** find a minimum $(k-1)$-way cut $\{Z_1, \ldots, Z_{k-1}\}$ in $G[V - X_i]$, and add the k-way cut $\{X_i, Z_1, \ldots, Z_{k-1}\}$ to $\mathcal{C}$;
6 if $|X_i| \geq k-1$ **then** find a minimum $(k-1)$-way cut $\{Z'_1, \ldots, Z'_{k-1}\}$ in $G[X_i]$, and add the k-way cut $\{Z'_1, \ldots, Z'_{k-1}, V - X_i\}$ to $\mathcal{C}$;
7 $i := i + 1$
8 **end**; /* while */
9 Let X_r be the last cut added to $\mathcal{X}$, and choose a cut $X_q \in \mathcal{X}$ that crosses X_r, where we denote $Y_1 = X_q - X_r$, $Y_2 = X_q \cap X_r$, $Y_3 = X_r - X_q$, and $Y_4 = V - (X_q \cup X_r)$;
10 **if** $k = 3$ **then** add to $\mathcal{C}$ 3-way cuts $\{Y_j, Y_{j+1}, V - (Y_j \cup Y_{j+1})\}$, $j \in \{1, 2, 3, 4\}$ (where $Y_{j+1} = Y_1$ for $j = 4$);
11 **if** $k = 4$ **then** add 4-way cut $\{Y_1, Y_2, Y_3, Y_4\}$ to $\mathcal{C}$;
12 Output a k-way cut π in $\mathcal{C}$ with the minimum weight.

The correctness of algorithm MULTIWAY follows from Theorem 2. To analyze the running time of MULTIWAY, let $C_k(n, m)$ $(k \geq 2)$ denote the time required to find a minimum k-way cut in an edge-weighted graph with n vertices and m edges.

By Lemma 1, the number r of iterations of the while-loop is at most $3n - 5 = O(n)$. Therefore, lines 5 and 6 in MULTIWAY requires $O(nC_{k-1}(n, m))$ time in total. The total time required to check whether $\mathcal{X}$ is crossing or not is $O(rn^2) = O(n^3)$. By Theorem 1, r minimum 2-way cuts can be enumerated in the non-decreasing order in $O(rnF(n, m)) = O(n^2F(n, m))$ time, where $F(n, m)$ denotes the time required to find a minimum (s, t)-cut (i.e., a minimum 2-way cut that separates two specified vertices s and t) in an edge-weighted graph with n vertices and m edges. Summing up these, we establish the following result.

Theorem 3 *For an edge-weighted graph $G = (V, E)$, where $n = |V| \geq 4$ and $m = |E|$, a minimum 3-way cut and a minimum 4-way cut can be computed in $O(n^2F(n, m) + nC_2(n, m) + n^3)$ and $O(n^2F(n, m) + nC_3(n, m) + n^3) = O(n^3F(n, m) + n^2C_2(n, m) + n^4)$ time, respectively.* □

For a planar graph G, we can obtain better bounds by applying Corollary 1.

Corollary 2 *For an edge-weighted planar graph $G = (V, E)$, where $n = |V| \geq 4$ and $m = |E|$, a minimum 3-way cut and a minimum 4-way cut can be computed in $O(n^3)$ and $O(n^4)$ time, respectively.* □

4 3-Way Cuts in Symmetric Submodular Systems

4.1 General case

Let (V, f) be a symmetric submodular system. Let us consider whether the algorithm MULTIWAY can be extended to find a minimum k-way cut in (V, f) for $k = 3, 4$. In the case of a graph $G = (V, E)$, if the first cut V_1 in a minimum k-way cut $\{V_1, \ldots, V_k\}$ is known, then the remaining task of finding cuts $V_2, \ldots, V_k$ can be reduced to computing a minimum $(k-1)$-way cut in the induced subgraph $G[V - V_1]$. However, even if the first cut V_1 in a minimum k-way cut $\{V_1, \ldots, V_k\}$ is known in (V, f), finding the rest of cuts $V_2, \ldots, V_k$ may not be directly reduced to the minimum $(k-1)$-way cut problem in a certain symmetric submodular system $(V - V_1, f')$ (because, we do not have any concept corresponding to the induced subgraph $G[V - V_1]$ in a symmetric submodular system (V, f)). Thus, we need to find a procedure to compute a minimum k-way cut $\{V_1, \ldots, V_k\}$ in (V, f) under the restriction that V_1 is fixed. In what follows, we show that such a procedure can be constructed for $k = 3$ (but the case of $k = 4$ is still open).

For this, we first define a set function f' on $V' = V - V_1$ from the cut V_1.

Lemma 3 *For a symmetric submodular system (V, f) and a subset $V_1 \subset V$, the set function f' on $V' = V - V_1$, defined by $f'(X) = f(X) + f(V' - X) - f(V')$, $X \subseteq V'$, is symmetric and submodular.*

Proof. Clearly f' is symmetric by definition. For any $X, Y \subseteq V$, we obtain $f'(X) + f'(Y) = f(X) + f(V' - X) - f'(V) + f(Y) + f(V' - Y) - f'(V) \geq f(X \cap Y) + f(X \cup Y) + f(V' - (X \cap Y)) + f(V' - (X \cup Y)) - 2f'(V) = f'(X \cap Y) + f'(X \cup Y)$. □

Note that for a 3-way cut $\pi = \{V_1, V_2, V_3\}$ in (V, f), its weight $\omega_f(\pi)$ is given by $\frac{1}{2}(f(V_1) + f(V_2) + f(V - V_1 - V_2)) = f(V_1) + \frac{1}{2}(f(V_2) + f(V' - V_2) - f(V_1)) = f(V_1) + \frac{1}{2}f'(V_2)$ for $V' = V - V_1$. Therefore, a minimum 3-way cut $\{V_1, V_2, V_3\}$ in (V, f) for a specified V_1 can be obtained by solving the minimum 2-way cut problem in the system (V', f').

For a symmetric submodular system (V, f), it is known that a minimum 2-way cut in a symmetric submodular system can be computed combinatorially in $O(n^3 T_f)$ time [18, 16], where T_f is the time to evaluate the function value $f(X)$ for a given subset $X \subseteq V$.

Now we are ready to extend the algorithm MULTIWAY so as to find a minimum 3-way cut in (V, f). By Theorem 1, we can enumerate 2-way cuts of (V, f) in the order of non-decreasing weights with $O(nF_f)$ time delay, where F_f is the time required to find a minimum 2-way cut that separates specified two disjoint subsets $S, T \subset V$ in (V, f). Thus, we can enumerate the first r minimum 2-way cuts of (V, f) in $O(rnF_f) = O(n^2 F_f)$ time, using the result $r = O(n)$ of Lemma 1. Therefore, we can compute a minimum 3-way cut $\{V_1, V_2, V_3\}$ in (V, f) under the restriction that V_1 is fixed. Summarizing these up, we have the following result.

Theorem 4 *For a symmetric and submodular system (V, f) with $n = |V| \geq 3$, a minimum 3-way cut can be computed in $O(n^2 F_f + n^4 T_f)$ time, where F_f is the time required to find a minimum 2-way cut that separates specified two disjoint subsets $S, T \subset V$ in (V, f), and T_f is the time to evaluate the function value $f(X)$ for a given subset $X \subseteq V$.* □

4.2 Cut function of a hypergraph

Now we elaborate upon the result of Theorem 4, assuming that f is the cut function c of a hypergraph. Let $H = (V, E)$ be an edge-weighted hypergraph with a vertex set V and a hyper-edge set E, where a hyper-edge e is defined as a non-empty subset of V. For a hyper-edge $e \in E$, $V(e)$ denotes the set of end vertices of e (i.e., the vertices incident to e). A *cut function* c in H is defined by $c(X) = \{w(e) \mid V(e) \cap X \neq \emptyset,\ V(e) \cap (V - X) \neq \emptyset\}$, $X \subseteq V$, where $w(e)$ is the weight of edge e. It is easy to observe that the cut function c in a hypergraph H is also symmetric and submodular. We define the weight $\omega_c(\pi)$ of a k-way cut $\{V_1, \ldots, V_k\}$ also by (4).

Remark: The weight $\omega_c(\pi)$ of a minimum 2-way cut π is equal to the minimum weight sum of edges whose removal makes the hypergraph disconnected. This is the same as the case of the cut function in a graph. For $k \geq 3$, however, $\omega_c(\pi)$ of a minimum k-way cut $\pi = \{V_1, \ldots, V_k\}$ may not mean the minimum weight sum γ_H of edges whose removal generates at least k components, where

$$\gamma_H = \min\{\gamma_H(\pi) \mid k\text{-way cut } \pi\}, \tag{5}$$

$$\gamma_H(\pi) = \{w(e) \mid e \in E,\ V(e) \not\subseteq V_i \text{ for all } V_i \in \pi\}.$$

This is different from the case of a graph. The reason is because the weight $w(e)$ of a hyper-edge e may contribute to $\omega_c(\pi)$ by more than 1 if e is incident to more than two subsets in π. □

Let $H = (V, E)$ be a hypergraph with $|E| \geq |V|$. Given two specified vertices $s, t \in V$, it is known [13] that a minimum 2-way cut separating s and t in H can be computed by solving a single maximum flow problem in the auxiliary digraph with $|V| + 2|E|$ vertices and $2d_H + 2|E|$ edges, where $d_H = \sum_{e \in E} |V(e)|$. Thus, a minimum 2-way cut separating two disjoint subsets $S, T \subset V$ can be computed in $F_c = F(|V| + 2|E|, 2d_H + 2|E|) = \tilde{O}(|E| d_H)$ time by computing a minimum 2-way cut separating s and t in the hypergraph obtained from H by contracting S and T into single vertices s and t, respectively.

For a given subset $X \subset V' = V - V_1$, the function value $f'(X) = c(X) + c(V' - X) - c(V_1)$ can be evaluated in $T_c = O(n + d_H)$ time. Hence, by Theorem 4, we can compute a minimum 3-way cut in $O(|V|^2 F_c + |V|^4 T_c) = \tilde{O}(|V|^2 |E| d_H + |V|^4 (|V| + d_H)) = \tilde{O}((|V|^4 + |V|^2 |E|) d_H)$ time.

The running time can be slightly improved as follows. Notice that for a 3-way cut $\pi = \{V_1, V_2, V_3\}$ in $H = (V, E)$, its weight $\omega_c(\pi)$ is written as

$$\omega_c(\pi) = \sum\{w(e) \mid e \in E,\ |\{i \mid V_i \cap V(e) \neq \emptyset\}| = 2\}$$

$$+\frac{3}{2}\sum\{w(e) \mid e \in E,\ |\{i \mid V_i \cap V(e) \neq \emptyset\}| = 3\}.$$

For a given cut V_1, let us define an edge-weighted hypergraph $H[V_1] = (V', E')$ induced from H by V_1 by $V' = V - V_1$ and $E' = \{e \in E \mid V(e) \cap (V - V_1) \neq \emptyset\}$, where for each hyper-edge $e \in E'$, the set $V'(e)$ of its end vertices and its weight $w'(e)$ are redefined by $V'(e) = V(e) - V_1$ and

$$w'(e) = \begin{cases} w(e) & \text{if } V'(e) = V(e) \\ w(e)/2 & \text{if } V'(e) \subset V(e). \end{cases}$$

Hence, for a 2-way cut $\pi' = \{V_2, V_3\}$ in $H[V_1] = (V', E')$, its weight $\omega_{c'}(\pi') = \sum\{w'(e) \mid e \in E',\ V'(e) \cap V_2 \neq \emptyset \neq V'(e) \cap V_3\}$ (where c' denotes the cut function in $H[V_1]$) is equal to $\sum\{w(e) \mid e \in E',\ V(e) \cap V_2 \neq \emptyset \neq V(e) \cap V_3,\ V(e) \cap V_1 = \emptyset\} + \sum\{w(e)/2 \mid e \in E',\ V'(e) \cap V_2 \neq \emptyset,\ V'(e) \cap V_3 \neq \emptyset,\ V'(e) \cap V_1 \neq \emptyset\}$. Thus, $c(V_1) + \omega_{c'}(\pi') = \omega_c(\pi)$ holds for a 3-way cut $\pi = \{V_1, V_2, V_3\}$ and the cut function c in H. Therefore, for a fixed cut V_1, we can find a minimum 3-way cut $\{V_1, V_2, V_3\}$ in H by computing a minimum 2-way cut $\{V_2, V_3\}$ in $H[V_1]$. It is known that a minimum 2-way cut in a hypergraph $H = (V, E)$ can be obtained in $O(|V|d_H + |V|^2 \log |V|)$ time [12]. Since we need to solve $O(n)$ number of minimum 2-way cut problems in our algorithm, the time complexity becomes $O(|V|^2 F(|V| + |E|, d_H) + |V|^2(d_H + |V| \log |V|)) = \tilde{O}(|V|^2|E|d_H)$, assuming $|E| \geq |V|$.

Corollary 3 *For an edge-weighted hypergraph $H = (V, E)$, a 3-way cut π minimizing $\omega_c(\pi)$ can be computed in $\tilde{O}(n^2 m d_H)$ time, where $n = |V|$, $m = |E| \geq n$, $d_H = \sum_{e \in E} |V(e)|$ and c is the cut function in H.* □

As remarked in the above, the weight of $\omega_c(\pi)$ of a minimum 3-way cut π in a hypergraph $H = (V, E)$ may not give the γ_H of (5). However, for any 3-way cut π, it holds $\omega_c(\pi) \leq \frac{3}{2}\gamma_H(\pi)$. Therefore, the minimum value $\omega_c(\pi)$ over all 3-way cuts π is at most 1.5 of γ_H.

Corollary 4 *For an edge-weighted hypergraph $H = (V, E)$, there is a 1.5-approximation algorithm for the problem of minimizing the sum of weights of hyper-edges whose removal leaves at least 3 components. The algorithm runs in $\tilde{O}(n^2 m d_H)$ time, where $n = |V|$, $m = |E|$ and $d_H = \sum_{e \in E} |V(e)| \geq n$.* □

For $k = 4$, it is not known how to find a minimum 4-way cut $\{V_1, \ldots, V_4\}$ in a symmetric submodular system (or in a hypergraph) under the condition that V_1 is fixed. By using the above argument, we can find a minimum 4-way cut in a hypergraph $H = (V, E)$ if every hyper-edge e has at most *three* incident vertices (i.e., $2 \leq |V(e)| \leq 3$ for all $e \in E$). This is because if $|V(e)| \leq 3$ for all $e \in E$, then it holds $\omega_c(\pi) = c(V_1) + \omega_{c'}(\pi')$ for the weight $\omega_c(\pi)$ of 4-way cut $\pi = \{V_1, V_2, V_3, V_4\}$ in H and the weight $\omega_{c'}(\pi')$ of a 3-way cut $\pi' = \{V_2, V_3, V_4\}$ in the induced hypergraph $H[V_1]$ as defined above. The algorithm runs in $\tilde{O}(|V|^2 F_c + |V|(|V|^2|E|d_H)) = \tilde{O}(|V|^3|E|d_H)$ time. Also, in this class of hypergraphs, a 4-way cut π that minimizes the weight $\omega_c(\pi)$ is a 1.5-approximation to the problem of finding a 4-way cut π that minimizes $\gamma_H(\pi)$.

Corollary 5 *Let $H = (V, E)$ be an edge-weighted hypergraph such that every hyper-edge $e \in E$ has at most three incident vertices, and let $n = |V|$, $m = |E| \geq n$ and $d_H = \sum_{e \in E} |V(e)|$. Then a 4-way cut π that minimizes $\omega_c(\pi)$ can be computed in $\tilde{O}(n^3 m d_H)$ time, and its $\gamma_H(\pi)$ value is at most 1.5 of the minimum weight sum γ_H of hyper-edges whose removal leaves at least 4 components in H.* □

5 Concluding Remark

In this paper, we proposed an algorithm for computing minimum 3-way and 4-way cuts in a graph, based on the enumeration algorithm of 2-way cuts in the non-decreasing order. The algorithm is then extended to the problem of finding a minimum 3-way cut in a symmetric submodular system. As a result of this, the minimum 3-way cut problem in a hypergraph can be solved in polynomial time. It is left for future work how to extend the algorithm to the case of $k \geq 5$ in graphs and the case of $k \geq 4$ in symmetric submodular systems (or in hypergraphs). Very recently, we proved that the minimum k-way cuts in graphs can be computed in $O(mn^k \log(n^2/m))$ time for $k = 5, 6$.

Acknowledgments

This research was partially supported by the Scientific Grant-in-Aid from Ministry of Education, Science, Sports and Culture of Japan, and the subsidy from the Inamori Foundation. We would like to thank Naoki Katoh and Satoru Iwata for their valuable comments.

References

[1] M. Burlet and O. Goldschmidt, *A new and improved algorithm for the 3-cut problem*, Operations Research Letters, vol.21, (1997), pp. 225–227.

[2] S. Even and R. E. Tarjan, *Network flow and testing graph connectivity*, SIAM J. Computing, vol.4, (1975), pp. 507–518.

[3] A. V. Goldberg and R. E. Tarjan, *A new approach to the maximum flow problem*, J. ACM, vol.35, (1988), pp. 921–940.

[4] M. Grötschel, L. Lovász and A. Schrijver, Geometric Algorithms and Combinatorial Optimization, Springer, Berlin (1988).

[5] O. Goldschmidt and D. S. Hochbaum, *A polynomial algorithm for the k-cut problem for fixed k*, Mathematics of Operations Research, vol.19, (1994), pp. 24–37.

[6] J. Hao and J. Orlin, *A faster algorithm for finding the minimum cut in a directed graph*, J. Algorithms, vol. 17, (1994), pp. 424–446.

[7] D. S. Hochbaum and D. B. Shmoys, *An $O(|V|^2)$ algorithm for the planar 3-cut problem*, SIAM J. Algebraic Discrete Methods, vol.6, (1985), pp. 707–712.

[8] Y. Kamidoi, S. Wakabayashi and N. Yoshida, *Faster algorithms for finding a minimum k-way cut in a weighted graph*, Proc. IEEE International Symposium on Circuits and Systems, (1997), pp. 1009–1012.

[9] Y. Kamidoi, S. Wakabayashi and N. Yoshida, *A new approach to the minimum k-way partition problem for weighted graphs*, Technical Report of Inst. Electron. Inform. Comm. Eng., COMP97-25, (1997), pp. 25–32.

[10] S. Kapoor, *On minimum 3-cuts and approximating k-cuts using cut trees*, Lecture Notes in Computer Science 1084, Springer-Verlag, (1996), pp. 132–146.

[11] D. R. Karger and C. Stein, *A new approach to the minimum cut problems*, J. ACM, vol.43, no.4, (1996), pp. 601–640.

[12] R. Klimmek and F. Wagner, *A simple hypergraph min cut algorithm*, Technical Report B 96-02, Department of Mathematics and Computer Science, Freie Universität Berlin (1996).

[13] E. L. Lawler, *Cutsets and partitions of hypergraphs*, Networks, vol.3, (1973), pp. 275–285.

[14] T. Lengaur, Combinatorial Algorithms for Integrated Circuit Layout, Wiley (1990).

[15] H. Nagamochi and T. Ibaraki, *Computing the edge-connectivity of multigraphs and capacitated graphs*, SIAM J. Discrete Mathematics, vol.5, (1992), pp. 54-66.

[16] H. Nagamochi and T. Ibaraki, *A note on minimizing submodular functions*, Information Processing Letters, vol.67, (1998), pp.239–244.

[17] H. Nagamochi, K. Nishimura and T. Ibaraki, *Computing all small cuts in undirected networks*, SIAM J. Discrete Mathematics, vol. 10, (1997), pp. 469–481.

[18] M. Queyranne, *Minimizing symmetric submodular functions*, Mathematical Programming, 82 (1998), pp. 3–12.

[19] V. V. Vazirani and M. Yannakakis, *Suboptimal cuts: Their enumeration, weight, and number*, Lecture Notes in Computer Science, 623, Springer-Verlag, (1992), pp. 366–377.