SIAM J. COMPUT.
Vol. 36, No. 5, pp. 1329–1341

A DETERMINISTIC ALGORITHM FOR FINDING ALL MINIMUM k-WAY CUTS*

YOKO KAMIDOI†, NORIYOSHI YOSHIDA†, AND HIROSHI NAGAMOCHI‡

Abstract. Let $G = (V, E)$ be an edge-weighted undirected graph with n vertices and m edges. We present a deterministic algorithm to compute a minimum k-way cut of G for a given k. Our algorithm is a divide-and-conquer method based on a procedure that reduces an instance of the minimum k-way cut problem to $O(n^{2k-5})$ instances of the minimum $(\lfloor (k + \sqrt{k})/2 \rfloor + 1)$-way cut problem, and can be implemented to run in $O(n^{4k/(1-1.71/\sqrt{k})-31})$ time. With a slight modification, the algorithm can find all minimum k-way cuts in $O(n^{4k/(1-1.71/\sqrt{k})-16})$ time.

Key words. minimum cut, multiway cut, divide-and-conquer

AMS subject classifications. 05C85, 68R10, 68W05

DOI. 10.1137/050631616

1. Introduction.

For an edge-weighted graph $G = (V, E)$, a subset F of edges is called a *k-way cut* if removal of F from G results in at least k connected components. The *minimum k-way cut problem* asks to find a minimum weight k-way cut in G. Given k vertices (called *terminals*), a k-way cut F is called a *k-terminal cut* if no two terminals are in the same connected component after removal of F. The problem of finding a minimum weight k-terminal cut is called the *minimum k-terminal cut problem*. These problems have several important applications such as VLSI design [1, 6, 26], task allocation in distributed computing systems [25, 34], graph strength [4, 9, 32], and network reliability [3, 35].

For $k = 2$, the minimum 2-terminal cut problem in a graph can be solved by applying a maximum flow algorithm. Let $F(n, m)$ denote the time complexity of a maximum flow algorithm in an edge-weighted graph with n vertices and m edges. The complexity $F(n, m)$ was found to be $O(n^3)$ in [24] and $O(nm \log(n^2/m))$ in [7]. Dahlhaus et al. [5] proved that the minimum k-terminal cut problem is NP-hard for any fixed $k \geq 3$. Several approximation algorithms have been proposed [2, 5, 21], among which a 1.3438-approximation algorithm is obtained by Karger et al. [21]. An extension of this problem to a general setting defined by submodular set functions can be found in the articles by Zhao, Nagamochi, and Ibaraki [39, 40]. For planar graphs, the minimum k-terminal cut problem admits a polynomial time algorithm [5], and currently an $O(k4^k n^{2k-4} \log n)$ time algorithm in [14] and an $O((k - \frac{3}{2})^{k-1}(n - k)^{2k-4}[nk - \frac{3}{2}k^2 + \frac{1}{2}k] \log(n - k))$ time algorithm in [37] are known.

On the other hand, Goldschmidt and Hochbaum [8] proved that the minimum k-way cut problem is NP-hard if k is an input parameter but admits a polynomial time algorithm if k is regarded as a constant. The minimum 2-way cut problem (i.e.,

*Received by the editors May 15, 2005; accepted for publication (in revised form) June 12, 2006; published electronically December 21, 2006. A preliminary version of this paper appeared in *Proceedings of the 4th Japanese–Hungarian Symposium on Discrete Mathematics and Its Applications*, Budapest, Hungary, 2005, pp. 224–233. This research was supported by the Scientific Grant-in-Aid from Ministry of Education, Science, Sports and Culture of Japan.

http://www.siam.org/journals/sicomp/36-5/63161.html

†Faculty of Information Sciences, Hiroshima City University, 3-4-1, Ozuka-Higashi, Asaminami-ku, Hiroshima 731-3194, Japan (yoko@ce.hiroshima-cu.ac.jp, nyoshida@khaki.plala.or.jp).

‡Department of Applied Mathematics and Physics, Kyoto University, Sakyo, Kyoto 606-8501, Japan (nag@amp.i.kyoto-u.ac.jp).

the problem of computing edge-connectivity) can be solved by $O(nm + n^2 \log n)$ and $O(nm \log(n^2/m))$ time deterministic algorithms [12, 28] and by $O(n^2(\log n)^3)$ and $O(m(\log n)^3)$ time randomized algorithms [20, 22, 23]. Approximation algorithms for a minimum k-way cut problem of G have been proposed in [16, 33, 38]. Saran and Vazirani [33] first proposed a $2(1-1/k)$-approximation algorithm for the minimum k-way cut problem, which runs in $O(nF(n, m))$ time. Kapoor [16] also gave a $2(1-1/k)$-approximation algorithm for the minimum k-way cut problem of G, which requires $O(k(nm+n^2 \log n))$ time. Zhao et al. [38] also presented an approximation algorithm by using a set of minimum 3-way cuts. Their algorithm has the performance ratio $2-3/k$ for an odd k and $2-(3k-4)/(k^2-k)$ for an even k, and runs in $O(kmn^3 \log(n^2/m))$ time. Approximation algorithms for a multiway cut problem defined by submodular set functions are discussed in the articles by Zhao, Nagamochi, and Ibaraki [39, 40].

Goldschmidt and Hochbaum [8] presented an $O(n^{k^2/2-3k/2+4}F(n, m))$ time algorithm for solving the minimum k-way cut problem. This running time is polynomial for any fixed k. The algorithm is based on a divide-and-conquer approach. Suppose that we can choose a family $\mathcal{X}$ of subsets of V such that at least one subset $X \in \mathcal{X}$ has a property that a minimum $(k-1)$-way cut $\{V_1, \ldots, V_{k-1}\}$ in the subgraph $G[V-X]$ induced by $V-X$ gives rise to a minimum k-way cut $\{X, V_1, \ldots, V_{k-1}\}$ in the original graph G. Then we can find a minimum k-way cut in G by solving the $(k-1)$-way cut problem instances $G[V-X]$ for all $X \in \mathcal{X}$. Goldschmidt and Hochbaum [8] proved that such a family $\mathcal{X}$ of $O(n^{2k-3})$ subsets of V can be found in polynomial time, implying an $O(n^{O(k^2)})$ time algorithm for the minimum k-way cut problem.

For small $k \le 6$ or planar graphs, faster algorithms have been obtained [16, 17, 18, 29, 30, 31]. For $k \le 6$, the above family $\mathcal{X}$ can be constructed in polynomial time by collecting $O(n)$ subsets of V, and an $O(mn^k \log(n^2/m))$ time algorithm is known [29, 30, 31]. For planar graphs, Hartvigsen [13] gave an $O(n^{2k-1})$ time algorithm, and Nagamochi and Ibaraki [30] and Nagamochi, Katayama, and Ibaraki [31] showed that the problem can be solved in $O(n^k)$ time if $k \le 6$. The case of unweighted planar graphs with $k = 3$ can be solved in $O(n \log n)$ time [15].

Randomized algorithms have been developed for the k-way cut problem. Karger and Stein [23] proposed a Monte Carlo algorithm for the minimum k-way cut problem which runs in $O(n^{2(k-1)} \log^3 n)$ time. Afterward, Levine [27] gave a Monte Carlo algorithm for $k \le 6$ that runs in $O(mn^{k-2} \log^3 n)$ time. However, for a general k and a general graph G, no faster deterministic algorithm has been discovered since Goldschmidt and Hochbaum [8] found an $O(n^{O(k^2)})$ time algorithm.

In this paper, we present the first $O(n^{O(k)})$ time deterministic algorithm to compute a minimum k-way cut of G. Our algorithm is based on a divide-and-conquer method which consists of a procedure that reduces an instance of the minimum k-way cut problem to $O(n^{2k-5})$ instances of the minimum $(\lfloor (k+\sqrt{k})/2 \rfloor + 1)$-way cut problem, and can be implemented to run in $O(n^{4k/(1-1.71/\sqrt{k})-31})$ time. With a slight modification, we can also find all minimum k-way cuts in $O(n^{4k/(1-1.71/\sqrt{k})-16})$ time.

The paper is organized as follows. Section 2 introduces notation and reviews basic properties of 2-way cuts. Section 3 presents our divide-and-conquer algorithm, assuming an efficient procedure for computing a family $\mathcal{X}$ of subsets required to reduce a given problem instance, which is discussed in section 5, after proving a key property on crossing 2-way cuts in section 4. Section 6 analyzes the runtime of our algorithm. Section 7 shows how to modify the algorithm so that all minimum k-way cuts can be computed, and section 8 makes some concluding remarks.

2. Preliminaries. Let $G = (V, E)$ stand for an edge-weighted undirected graph consisting of a vertex set V and an edge set E with an edge weight function $cost : E \to R^+$, where R^+ is the set of nonnegative real numbers. Let $n = |V|$ be the number of vertices and $m = |E|$ be the number of edges. We may simply call G a graph. Let $comp(G)$ denote the number of connected components in G. An edge $e \in E$ with end vertices u and v may be denoted by $e = (u, v)$, and its weight is denoted by $cost(e)$. For a nonempty subset $F \subseteq E$, we let $cost(F)$ denote $\sum_{e \in F} cost(e)$. Let $X_1, X_2, \ldots, X_p$ be mutually disjoint subsets of V.

We denote the set of edges $e = (u, v)$ with $u \in X_i$ and $v \in X_j$ for some $i \neq j$ by $(X_1; X_2; \ldots; X_p)$, and the sum of the weights of these edges by $cost(X_1; X_2; \ldots; X_p)$, which is defined to be 0 if $(X_1; X_2; \ldots; X_p) = \emptyset$. For a subset X of V, we may denote $f(X) = cost(X; V-X)$, where f is called a *cut function* of G and satisfies the following identities:

$$f(X) + f(Y) = f(X \cap Y) + f(X \cup Y) + 2cost(X-Y, Y-X) \quad \text{for all } X, Y \subseteq V, \tag{2.1}$$

$$f(X) + f(Y) = f(X-Y) + f(Y-X) + 2cost(X \cap Y, V-(X \cup Y)) \quad \text{for all } X, Y \subseteq V. \tag{2.2}$$

Let F be a subset of E in G. We denote by $G - F$ the graph obtained from G by deleting edges in F. We call F a *k-way cut* if $comp(G - F) \geq k$. A k-way cut F is *minimum* if it has the minimum $cost(F)$ over all k-way cuts. Given a graph G and an integer $k (\geq 2)$, the *minimum k-way cut problem* asks to find a minimum k-way cut in G. We denote the cost of a minimum k-way cut in G by $opt(G, k)$. Note that $opt(G, k) = 0$ if and only if the set of edges with positive weights induces a subgraph of G with at least k connected components. Any inclusionwise minimal k-way cut F is given by $F = (X_1; X_2; \ldots; X_k)$ for some partition $\{X_1, X_2, \ldots, X_k\}$ of V. Conversely, for any partition $\{V_1, V_2, \ldots, V_k\}$ of V, $F' = (V_1; V_2; \ldots; V_k)$ is a k-way cut, where possibly $comp(G-F') > k$. For a set $\mathcal{C}$ of subsets F of E, we denote the union $\cup_{F \in \mathcal{C}} F$ by $E(\mathcal{C})$.

Given a nonempty vertex subset X, let $G[X] = (X, E_X)$ be the subgraph of G induced by X, where $G[X]$ has the edge weight function $cost_X : E_X \to R^+$, which is defined such that $cost_X(e) = cost(e)$ for every edge $e \in E_X$. For a subset Y of vertices of V, we denote $V - Y$ by $\overline{Y}$ if V is clear from the context.

Given p mutually disjoint nonempty subsets $T_1, T_2, \ldots, T_p$ of V, called *terminal sets*, a subset $F \subseteq E$ is called a *$(T_1, T_2, \ldots, T_p)$-terminal cut* of G if the removal of F from G disconnects each terminal set from the others. A $(T_1, T_2, \ldots, T_p)$-terminal cut is called minimum if it has the minimum $cost(F)$ among all $(T_1, T_2, \ldots, T_p)$-terminal cuts.

3. Divide-and-conquer algorithm. Each of the previously known deterministic algorithms reduces a minimum k-way cut problem instance to a set of minimum $(k-1)$-way cut problem instances, where the target k on the number of components is reduced only by 1. In this paper, we reduce a minimum k-way cut problem instance to a set of minimum k'-way cut problem instances with k' nearly equal to $k/2$. For this, we first observe the following property.

LEMMA 3.1. *Let $(V_1; V_2; \ldots; V_k)$ be a minimum k-way cut in a graph $G = (V, E)$, where $k \in [2, n]$.*

Then for any integer $p \in [1, k-1]$*, there is a union* X *of* p *subsets in* $\{V_1, V_2, \ldots, V_k\}$ *such that*

$$f(X) \leq \frac{2(kp - p^2)}{(k^2 - k)} opt(G, k).$$

Proof. Let $\mathcal{X}$ be the family of all such unions X. Then $|\mathcal{X}| = \binom{k}{p}$. For each edge $e = (u, v) \in (V_1; V_2; \ldots; V_k)$, there are $2\binom{k-2}{p-1}$ unions $X \in \mathcal{X}$ such that $u \in X$ and $v \in \overline{X}$ or $u \in \overline{X}$ and $v \in X$. Therefore it holds that $\sum_{X \in \mathcal{X}} f(X) = 2\binom{k-2}{p-1} opt(G, k)$, and the average of $f(X)$ over all $X \in \mathcal{X}$ is $[2\binom{k-2}{p-1} opt(G, k)]/\binom{k}{p} = 2(kp - p^2)/(k^2 - k) opt(G, k)$. This implies the lemma. □

Let $p = \lceil (k - \sqrt{k})/2 \rceil - 1$, which satisfies $2(kp - p^2)/(k^2 - k) < 1/2$ for $k \geq 5$. Then there exists a set $X \subseteq V$ such that

$$f(X) < opt(G, k)/2$$

and X is a union of p subsets in $\{V_1, V_2, \ldots, V_k\}$ for a minimum k-way cut $(V_1; V_2; \ldots; V_k)$ in G. For such a subset X, we can reduce the current instance (G, k) into two instances $(G[X], p)$ and $G([V - X], k - p)$, where a minimum k-way cut F for (G, k) is obtained from a minimum p-way cut F' for $(G[X], p)$ and a minimum $(k - p)$-way cut F'' for $(G[V - X], k - p)$ by constructing a k-way cut $F = (X; V - X) \cup F' \cup F''$. Note that the size k is reduced to at most $k - \lceil (k - \sqrt{k})/2 \rceil + 1 = \lfloor (k + \sqrt{k})/2 \rfloor + 1$, which is nearly a half of k for a large k.

Section 5 shows that the number of such subsets $X \in V$ with $f(X) < opt(G, k)/2$ is at most n^{2k-5} and a family $\mathcal{X}$ of n^{2k-5} subsets including these subsets X (possibly together with some other subsets) can be obtained in $O(n^{2k-5} F(n, m))$ time. With this property, our divide-and-conquer algorithm can be described as follows.

Algorithm MULTIWAY(G, k)
Input: A graph $G = (V, E)$ and an integer $k \in [1, |V|]$.
Output: A minimum k-way cut F in G.
1. **if** $opt(G, k) = 0$ **then** Return $F := \emptyset$
2. **else** /* $opt(G, k) > 0$ */
3.   **if** $k \leq 2$ **then** Return a minimum k-way cut F of G in the time of $O(1)$ maximum flow computations
4.   **else if** $k \leq 6$ **then** Return a minimum k-way cut F of G by $O(|V|^{k-1})$ maximum flow computations
5.   **else** /* $k \geq 7$ */
6.   Compute a set $\mathcal{X}$ of at most $|V|^{2k-5}$ subsets of V such that any 2-way cut with cost less than $opt(G, k)/2$ is given by $(X; V - X)$ for some $X \in \mathcal{X}$;
7. $p := \lceil (k - \sqrt{k})/2 \rceil - 1$;
8.   **for** each $X \in \mathcal{X}$ with $|X| \geq p$ and $|V - X| \geq k - p$ **do**
9. $F_X := (X; V - X) \cup \mathrm{MULTIWAY}(G[X], p) \cup \mathrm{MULTIWAY}(G[V - X], k - p)$;
10.   **end** /* for */
11.   Choose a k-way cut F_X with the minimum cost over all X, and return $F := F_X$
12.   **end** /* if */
13. **end**. /* if */

For the correctness of algorithm MULTIWAY, we have only to give a procedure in line 5, which will be discussed in section 5. The runtime of MULTIWAY will be analyzed in section 6.

4. A crossing property. This section provides a property on crossing 2-way cuts, based on which a procedure for collecting all subsets $X \subset V$ with $f(X) < opt(G,k)/2$ is designed in section 5.

LEMMA 4.1. *For a graph $G = (V,E)$, let $\{Y_1, Y_2, \ldots, Y_q, W, Z\}$ be a partition of V, and let Q be a subset of V such that each subset in $\{Y_1 - Q, Y_2 - Q, \ldots, Y_q - Q, W \cap Q, Q \cap Z, Z - Q\}$ is nonempty. Then partition $\{Y'_i = Y_i - Q$ $(i = 1, 2, \ldots, q)$, $Y'_{q+1} = Q \cap Z$, $W' = (W \cup Q) - Z$, $Z' = Z - Q\}$ of V satisfies*

$$\begin{aligned} &2cost(Y_1;Y_2;\ldots;Y_q;W;Z) - f(Y_{1,q}) + f(Q) \\ &\quad \geq 2cost(Y'_1;Y'_2;\ldots;Y'_q;Y'_{q+1};W';Z') - f(Y'_{1,q+1}) + f(W \cap Q), \end{aligned}$$

where we denote $Y_{1,i} = Y_1 \cup Y_2 \cup \cdots \cup Y_i$ and $Y'_{1,j} = Y'_1 \cup Y'_2 \cup \cdots \cup Y'_j$.

Proof. We obtain

$$\begin{aligned} &f(Y_1) + f(Y_2) + \cdots + f(Y_q) + f(Y'_{1,q+1}) \\ &\quad \geq f(Y'_1) + f(Y'_2) + \cdots + f(Y'_q) + f(Y_{1,q} \cup (Q \cap Z)) + 2cost(Y_{1,q} \cap Q; Q \cap Z), \end{aligned} \tag{4.1}$$

by summing up the following q inequalities implied by (2.1):

$$\begin{aligned} &f(Y_1) + f((Y_{1,q} - Q) \cup (Q \cap Z)) \\ &\quad \geq f(Y_1 - Q) + f(Y_1 \cup (Y_{1,q} - Q) \cup (Q \cap Z)) + 2cost(Y_1 \cap Q; Q \cap Z), \\ &f(Y_2) + f(Y_1 \cup (Y_{1,q} - Q) \cup (Q \cap Z)) \\ &\quad \geq f(Y_2 - Q) + f(Y_{1,2} \cup (Y_{1,q} - Q) \cup (Q \cap Z)) + 2cost(Y_2 \cap Q; Q \cap Z) \\ &\qquad\qquad \cdots \\ &f(Y_q) + f(Y_{1,q-1} \cup (Y_{1,q} - Q) \cup (Q \cap Z)) \\ &\quad \geq f(Y_q - Q) + f(Y_{1,q} \cup (Y_{1,q} - Q) \cup (Q \cap Z)) + 2cost(Y_q \cap Q; Q \cap Z). \end{aligned}$$

On the other hand, (2.1) and (2.2) mean

$$\begin{aligned} f(Z) + f(W) + f(Q) &\geq f(Z) + f(W \cap Q) + f(W \cup Q) \\ &\geq f((W \cup Q) - Z) + f(Z - Q) + 2cost(Y_{1,q} - Q; Q \cap Z) + f(W \cap Q). \end{aligned} \tag{4.2}$$

From (4.1) and (4.2), we have

$$\begin{aligned} &f(Y_1) + f(Y_2) + \cdots + f(Y_q) + f(W) + f(Z) + f(Y'_{1,q+1}) + f(Q) \\ &\geq f(Y'_1) + f(Y'_2) + \cdots + f(Y'_q) + f((W \cup Q) - Z) + f(Z - Q) \\ &\quad + f(Y_{1,q} \cup (Q \cap Z)) + 2cost(Y_{1,q} \cap Q; Q \cap Z) + 2cost(Y_{1,q} - Q; Q \cap Z) + f(W \cap Q) \\ &= f(Y'_1) + f(Y'_2) + \cdots + f(Y'_q) + f(W') + f(Z') + f(Q \cap Z) + f(Y_{1,q}) + f(W \cap Q), \end{aligned}$$

implying the lemma. ∎

LEMMA 4.2. *For a graph $G = (V,E)$ and an integer $k \in [5, n-1]$, let $(X; \overline{X})$ be a 2-way cut of G, R be a nonempty subset of $\overline{X}$, and $T = \{t_1, t_2, \ldots, t_p\}$ be a set of $p \geq 2$ vertices in $\overline{X} - R$. Assume that, for each t_i, there exists a minimum $(X, T \cup R - \{t_i\})$-terminal cut $(X_i; \overline{X_i})$ which satisfies $X \cup \{t_i\} \subseteq X_i$ (see Figure 4.1). Let $\mathcal{C} = \{(X_i; \overline{X_i}) \mid 1 \leq i \leq p\}$. Then $E(\mathcal{C})$ is a $(p+2)$-way cut which partitions V into $p+2$ subsets*

$$Z = \cap_{1 \leq i \leq p} \overline{X_i}, \quad W = \cup_{1 \leq i < j \leq p} (X_i \cap X_j), \text{ and } Y_i = X_i - W \ (i = 1, 2, \ldots, p).$$

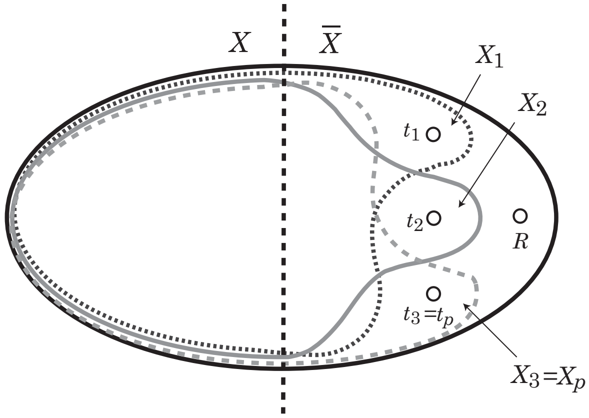

FIG. 4.1. *Illustration for a 2-way cut* $(X;\overline{X})$ *and a minimum* $(X;T\cup R-\{t_i\})$*-terminal cut* $(X_i;\overline{X_i})$*,* $i=1,2,3\ (=p)$*.*

Furthermore $(p+2)$*-way cut* $E(\mathcal{C})=(Y_1;Y_2;\ldots;Y_p;Z;W)$ *satisfies*

$$cost(E(\mathcal{C}))+cost(Z;W)+cost(Y_1;Y_2;\ldots;Y_p)\le f(X_1)+f(X_2). \tag{4.3}$$

Proof. Since $X\subseteq W$, $t_i\in Y_i$ $(1\le i\le p)$, and $R\subseteq Z$ hold, we see that $E(\mathcal{C})$ is a $(p+2)$-way cut. We prove (4.3) by an induction on p.

Basis case. For $p=2$, $Z=V-(X_1\cup X_2)$, $W=X_1\cap X_2$, $Y_1=X_1-X_2$, and $Y_2=X_2-X_1$. Then it holds that $cost(Y_1;Y_2;Z;W)+cost(Z;W)+cost(Y_1;Y_2)$ $=cost(X_1-X_2;X_2-X_1;V-(X_1\cup X_2);X_1\cap X_2)+cost(V-(X_1\cup X_2);X_1\cap X_2)$ $+cost(X_1-X_2;X_2-X_1)=f(X_1)+f(X_2)$, as required.

Inductive case. Let $q\ge 2$. Assuming that (4.3) holds for $p=q$, we prove that (4.3) holds for $p=q+1$. Let R', $T'=\{t_1,t_2,\ldots,t_{q+1}\}\subset\overline{X}-R'$ and $\mathcal{C}'$ be subsets of $\overline{X}$ and a set of $q+1$ 2-way cuts satisfying the condition of the lemma for $q+1$. We here consider $R=R'\cup\{t_{p+1}\}$, $T=T'-\{t_{q+1}\}$, and $\mathcal{C}=\mathcal{C}'-\{(X_{q+1};\overline{X_{q+1}})\}$, which satisfy the condition of the lemma for q (see Figure 4.2). Hence, by the induction

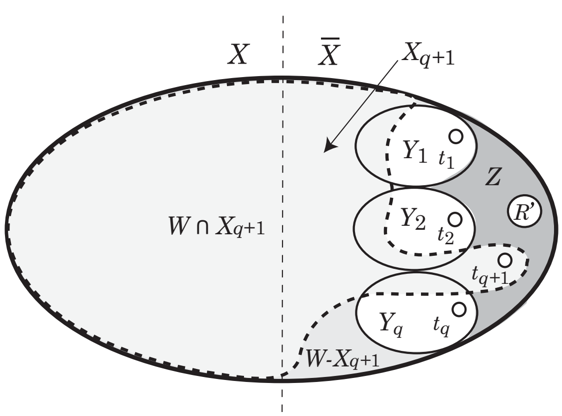

FIG. 4.2. *Illustration for a minimum* $(X,T'\cup R'-\{t_p\})$*-terminal cut* $(X_p;\overline{X_p})$ *and subsets* $Y_1,Y_2,\ldots,Y_{p-1}$*.*

hypothesis, we have

$$\begin{aligned} & f(X_1) + f(X_2) \\ & \quad \geq cost(E(\mathcal{C})) + cost(Z; W) + cost(Y_1; Y_2; \ldots; Y_q) \\ & \quad = 2cost(E(\mathcal{C})) - f(Y_{1,q}), \end{aligned} \tag{4.4}$$

where $Z = \cap_{1 \leq i \leq q} \overline{X_i}$, $W = \cup_{1 \leq i < j \leq q}(X_i \cap X_j)$, and $Y_i = X_i - W$ $(i = 1, 2, \ldots, q)$. By Lemma 4.1 to $Y_1, \ldots, Y_q$, W, Z, and $Q = X_{q+1}$, we obtain

$$\begin{aligned} 2cost(E(\mathcal{C})) - f(Y_{1,q}) &\geq 2cost(E(\mathcal{C}')) - f(Y'_{1,q+1}) + f(W \cap X_{q+1}) - f(X_{q+1}) \\ &\geq 2cost(E(\mathcal{C}')) - f(Y_{1,q+1}), \end{aligned}$$

where $f(W \cap X_{q+1}) \geq f(X_{q+1})$ holds since $(W \cap X_{q+1}; \overline{W \cap X_{q+1}})$ is an $(X, T' \cup R' - \{t_{q+1}\})$-terminal cut in G. This implies that (4.3) holds for p. $\square$

5. Computing small cuts. With Lemma 4.2, we are ready to present an $O(n^{2k-5}F(n,m))$ time procedure for collecting all subsets $X \in V$ with $f(X) < opt(G,k)/2$.

THEOREM 5.1. *For a graph $G = (V, E)$ and an integer $k \in [5, n-1]$, let $(X; \overline{X})$ be a 2-way cut with $f(X) < opt(G,k)/2$. Then, for any vertices $s^* \in X$ and $t^* \in \overline{X}$, there are subsets $S \subseteq X$ and $T \subseteq \overline{X}$ with $|S| \leq k-3$ and $|T| \leq k-3$ such that $(X; \overline{X})$ is a unique minimum $(S \cup \{s^*\}, T \cup \{t^*\})$-terminal cut in G.*

Proof. Let $(X; \overline{X})$ be a 2-way cut with $f(X) \leq opt(G,k)/2$. We first prove the next claim.

CLAIM 5.2. *A set T of at most $k-3$ vertices in $\overline{X}$ can be chosen so that $(X; \overline{X})$ becomes a unique minimum $(X, T \cup \{t^*\})$-terminal cut.*

Proof. Let $\mathcal{Y}$ be the family of all subsets Y with $X \subset Y \subseteq V - \{t^*\}$ such that

$$f(Y) \leq f(X).$$

We choose a subset T of $\overline{X} - \{t^*\}$ so that T becomes a minimal transversal of $\mathcal{Y}$ (i.e., T is an inclusion-wise minimal subset of $\overline{X} - \{t^*\}$ such that $Y \cap T \neq \emptyset$ for all $Y \in \mathcal{Y}$). Since T is a transversal of $\mathcal{Y}$, no other $(X, T \cup \{t^*\})$-terminal cut than $(X; \overline{X})$ has cost less than or equal to $f(X)$. Hence, to prove the claim, it suffices to show that $|T| \leq k-3$.

For each $t \in T$, let $(X_t; \overline{X}_t)$ denote a minimum $(X, (T - \{t\}) \cup \{t^*\})$-terminal cut. We show that

$$f(X_t) \leq f(X), \quad t \in X_t.$$

By the minimality of T, each vertex $t \in T$ has a subset $Y' \in \mathcal{Y}$ such that $t \in Y'$ and $Y' \cap (T - \{t\}) \cup \{t^*\}) = \emptyset$. Hence $f(X_t) \leq f(Y') \leq f(X)$. This also implies that $X_t \in \mathcal{Y}$ and hence t must belong to X_t (since otherwise $X_t \cap T = \emptyset$ would hold). See Figure 4.1, where $R = \{t^*\}$.

The above sets $R = \{t^*\}$, T, $\mathcal{C} = \{(X_t; \overline{X}_t) \mid t \in T\}$ satisfy the condition of Lemma 4.2. By Lemma 4.2 and the assumption on $f(X)$, $E(\mathcal{C})$ is a $(|T|+2)$-way cut with $cost(E(\mathcal{C})) \leq 2\max\{f(X_t) \mid t \in T\} \leq 2f(X) < opt(G,k)$. Therefore $|T| \leq k-3$ holds, since otherwise $comp(G - E(\mathcal{C})) \geq |T| + 2 \geq k$ would hold, contradicting the definition of $opt(G,k)$. This proves the claim. $\square$

By applying the above claim to X, we see that a set S of at most $k-3$ vertices in X can be chosen so that $(X; \overline{X})$ becomes a unique minimum $(S \cup \{s^*\}, \overline{X})$-terminal cut.

Finally we show that $(X;\overline{X})$ is a unique minimum $(S \cup \{s^*\}, T \cup \{t^*\})$-terminal cut in G. Assume indirectly that G has another minimum $(S\cup\{s^*\}, T\cup\{t^*\})$-terminal cut $(Z;\overline{Z})$. By the property of S and T, neither $Z \subseteq X$ nor $Z \supseteq X$; the remaining case is $X - Z \neq \emptyset \neq Z - X$. In this case, by the submodularity of cost function,

$$f(X) + f(Z) \geq f(X \cap Z) + f(Z \cup X)$$

holds, and we see that at least one of $(X \cap Z; \overline{X \cap Z})$ and $(Z \cup X; \overline{Z \cup X})$ is a minimum $(S \cup \{s^*\}, T \cup \{t^*\})$-terminal cut. This, however, contradicts the above property of S and T. This completes the proof of the theorem. ∎

Based on this theorem, we can find all 2-way cuts $(X;\overline{X})$ with $f(X) < opt(G,k)/2$ by $O(n^{2k-5})$ maximum flow computations. For this, choose a vertex $s^* \in V$, and execute the following procedure for each vertex $t^* \in V - \{s^*\}$: Choose disjoint sets $S, T \subseteq V - \{s^*, t^*\}$ with $2 \leq |S| \leq k-3$ and $2 \leq |T| \leq k-3$, and compute a minimum $(S \cup \{s^*\}, T \cup \{t^*\})$-terminal cut $(X;\overline{X})$ in G. Then the set of these 2-way cuts $(X;\overline{X})$ for all $t^* \in V - \{s^*\}$ include those with cost less than $opt(G,k)/2$. For fixed s^* and t^*, there are at most n^{2k-6} such pairs of S and T. Hence, we need $O(n^{2k-6} \cdot n) = O(n^{2k-5})$ maximum flow computations. We also have the following corollary.

Corollary 5.3. *Let $G = (V,E)$ be a graph, $k \in [5,n]$ be an integer, and $opt(G,k) > 0$. Then the number of subsets $X \subset V$ such that $f(X) < (1/2)opt(G,k)$ is $O(n^{2k-5})$.*

6. Runtime of MULTIWAY.

In this section, we analyze the runtime of algorithm MULTIWAY.

Theorem 6.1. *For a graph $G = (V,E)$ with n vertices, m edges, and an integer $k \in [1,n]$,* MULTIWAY(G,k) *runs in $O(n^{k-1}F(n,m))$ time for $k \leq 6$ and in $O(n^{4k/(1-1.71/\sqrt{k})-34}F(n,m))$ time for $k \geq 7$, where $F(n,m)$ denotes the time complexity for computing a maximum flow in a graph with n vertices and m edges.*

Proof. We derive an upper bound $N(k,n)$ on the number of maximum flow computations to execute MULTIWAY(G,k) for a graph G with n vertices, where we assume that $N(k,n)$ is an increasing function with respect to k and n. For $k \leq 6$, it is known that a minimum k-way cut can be obtained by at most 1 (resp., $2n^2$, $4n^3$, $60n^4$, and $900n^5$) maximum flow computations for $k = 2$ [12] (resp., $k = 3,4,5,6$ [29, 30, 31]). For $k \geq 5$, it suffices to consider $N(k,n)$ such that

$$\begin{aligned} N(k,n) &\leq n^{2k-5} + n^{2k-5}N(k - \lceil (k-\sqrt{k})/2 \rceil + 1, n - \lceil (k-\sqrt{k})/2 \rceil + 1) \\ &\leq n^{2k-5}N(\lfloor (k+\sqrt{k})/2 \rfloor + 1, n). \end{aligned}$$

Then we define $M(k)$ by a recursive formula $M(k) = 2k - 5 + M(\lfloor (k+\sqrt{k})/2 \rfloor + 1)$ for $k \geq 7$ and $M(2) = 0$, $M(3) = 2$, $M(4) = 3$, $M(5) = 4$, and $M(6) = 5$. We see that $900n^{M(k)}$ gives an upper bound on the number of maximum flow computations needed to execute MULTIWAY(G,k). We see that $M(k) \leq 4k/(1 - 1.71/\sqrt{k}) - 34$ holds for $k \leq 1.3 \times 10^6$ by generating all those $M(k)$ with a computer program. We prove that $M(k) \leq 4k/(1 - 1.71/\sqrt{k}) - 34$ holds for $k > 1.3 \times 10^6$ with the recursive formula. Let $a = 1.71$, and $k' = \lfloor (k+\sqrt{k})/2 \rfloor + 1 \leq (k + \sqrt{k} + 2)/2$. Then by the induction hypothesis we have

$$M(k) = 2k - 5 + M(k') \leq 2k - 5 + 4k'\sqrt{k'}/(\sqrt{k'} - a) - 34.$$

Then it suffices to show that

$$4k\sqrt{k}/(\sqrt{k}-a) - 34 - (2k-5) - 2(k+\sqrt{k}+2)\sqrt{k+\sqrt{k}+2}/(\sqrt{k+\sqrt{k}+2} - \sqrt{2}a) + 34$$

is nonnegative for $k > 1.3 \times 10^6$. For this, we prove the following is nonnegative:

$$\begin{aligned}
&4k\sqrt{k}(\sqrt{k+\sqrt{k}+2}-\sqrt{2}a)-(\sqrt{k}-a)(2k-5)(\sqrt{k+\sqrt{k}+2}-\sqrt{2}a)\\
&\quad -(\sqrt{k}-a)2(k+\sqrt{k}+2)\sqrt{k+\sqrt{k}+2}\\
&\quad =\left((4a-2)k+(2a+1)\sqrt{k}-a\right)\sqrt{k+\sqrt{k}+2}\\
&\quad +\sqrt{2}a(-2k\sqrt{k}-2ak-5\sqrt{k}+5a).
\end{aligned}\tag{6.1}$$

Since $(4a-2)k+(2a+1)\sqrt{k}-a > 0$ for $k > 1.3 \times 10^6$, (6.1) is at least

$$\begin{aligned}
&((4a-2)k+(2a+1)\sqrt{k}-a)\sqrt{k}+\sqrt{2}a(-2k\sqrt{k}-2ak-5\sqrt{k})\\
&\quad =\sqrt{k}\left[((4-2\sqrt{2})a-2)k+(2a(1-\sqrt{2}a)+1)\sqrt{k}-(1+5\sqrt{2})a\right],
\end{aligned}$$

which is nonnegative, since $4a-2\sqrt{2}a-2>0$ holds for $k = 1.3 \times 10^6$ and in this case it holds that

$$\begin{aligned}
((4-2\sqrt{2})\times 1.71-2)\times 1.3\times 10^6+(2\cdot 1.71\times(1-\sqrt{2}\cdot 1.71)+1)\times\sqrt{1.3\times 10^6}\\
-(1+5\sqrt{2})\times 1.71 > 0.
\end{aligned}$$

This proves that $M(k) \le 4k/(1-1.71/\sqrt{k})-34$ holds for $k > 1.3 \times 10^6$. ∎

7. Enumerating all minimum k-way cuts.

In this section, we show that algorithm MULTIWAY can be modified so that all minimum k-way cuts in G can be enumerated in nearly the same time complexity. Given a graph $G=(V,E)$ and integer k, we can construct all minimum k-way cuts F by combining a minimum p-way cut F' in $G[X]$ and a minimum $(k-p)$-way cut F'' in $G[V-X]$ for all possible subsets $X \subset V$ such that X is a union of p subsets in $\{V_1,\dots,V_k\}$ for a minimum k-way cut in G.

However, we cannot apply Corollary 5.3 to instances (G,k) with $k \le 4$. From Lemma 3.1 with $p=1$, we see that for any minimum k-way cut $(V_1;V_2;\dots;V_k)$ in a graph $G=(V,E)$, there is a subset $X \in \{V_1,V_2,\dots,V_k\}$ such that

$$f(X) \le (2/k)opt(G,k).$$

For $k \in \{2,3,4\}$, we derive an upper bound on the number of subsets $X \subset V$ with $f(X) \le (2/k)opt(G,k)$.

LEMMA 7.1. *Let $G=(V,E)$ be a graph, $k \in [2,4]$ be an integer, and $opt(G,k)>0$. Then the number of subsets $X \subset V$ such that $f(X) \le (2/k)opt(G,k)$ for $k=2$ (resp., $k=3,4$) is $O(n^2)$, (resp., $O(n^4)$ and $O(n^4)$).*

Proof. Let $\mathcal{X}_k$ be the family of subsets X with $f(X) \le (2/k)opt(G,k)$. Let $\lambda(G)$ denote the edge-connectivity of G (i.e., $\lambda(G)=opt(G,2)$). It is known [19] that, for $\lambda(G)>0$, there are $O(n^{2\alpha})$ subsets X with $f(X) \le \alpha\lambda(G)$.

(i) Let $k=2$. Then $\lambda(G)>0$ holds by assumption $opt(G,2)>0$. This proves that $|\mathcal{X}_2| = O(n^{2\alpha}) = O(n^2)$ holds for $\alpha=1$.

For $k \in \{3,4\}$, we assume that $\mathcal{X}_k$ contains two subsets X_1 and X_2 such that each of $Y_1=X_1\cap X_2$, $Y_2=X_1-X_2$, $Y_3=V-(X_1\cup X_2)$, and $Y_4=X_2-X_1$ is nonempty (otherwise $|\mathcal{X}_k| \le 2n$ holds).

(ii) Let $k = 3$. If $\lambda(G) = 0$, then the edges with positive weights induce from G exactly two connected components G_1 and G_2, where we see that a minimum 3-way cut in G is a minimum 2-way cut in G_1 or G_2. By the result of (i) we have $|\mathcal{X}_k| = O(n_1^2 + n_2^2) = O(n^2)$, where n_i is the number of vertices in G_i. We next assume $\lambda(G) > 0$. Then for the above two crossing subsets $X_1, X_2 \in \mathcal{X}_3$, we have

$$\begin{aligned}
4opt(G,3) &\leq cost(Y_1; Y_2; V - (Y_1 \cup Y_2)) + cost(Y_2; Y_3; V - (Y_2 \cup Y_3)) \\
&\quad + cost(Y_3; Y_4; V - (Y_3 \cup Y_4)) + cost(Y_4; Y_1; V - (Y_4 \cup Y_1)) \\
&= 3(f(X_1) + f(X_2)) - 2cost(Y_1; Y_3) - 2cost(Y_2; Y_4) \\
&\leq 3((2/3)opt(G,3) + (2/3)opt(G,3)) - 2cost(Y_1; Y_3) - 2cost(Y_2; Y_4) \\
&= 4opt(G,3) - 2cost(Y_1; Y_3) - 2cost(Y_2; Y_4),
\end{aligned}$$

implying that $f(X_1) = f(X_2) = (2/3)opt(G,3)$ and $cost(Y_1; Y_3) = cost(Y_2; Y_4) = 0$. Hence no two subsets $X, X' \in \mathcal{X}_3$ with $f(X) < (2/3)opt(G,3)$ and $f(X') < (2/3)opt(G,3)$ cross each other, indicating that the number of subsets X with $f(X) < (2/3)opt(G,3)$ is $O(n)$. If $\lambda(G) < (1/3)opt(G,3)$, then there is a subset $Z \subset V$ with $f(Z) = \lambda(G) < (1/3)opt(G,3) (< f(X_1))$, where $Z \neq X_1 \neq V - Z$ holds, and $F = (Z; V-Z) \cup (X_1; V-X_1)$ is a 3-way cut with $cost(F) < (1/3)opt(G,3) + (2/3)opt(G,3) = opt(G,3)$, which is a contradiction. Therefore, $\lambda(G) \geq (1/3)opt(G,3)$ holds, and $f(X) = (2/3)opt(G,3) \leq 2\lambda(G)$ holds for every subset X with $f(X) = (2/3)opt(G,3)$, implying that $|\mathcal{X}_3| = O(n^{2\alpha}) = O(n^4)$ holds for $\alpha = 2$.

(iii) Let $k = 4$. If $\lambda(G) = 0$, then it is not difficult to see that $|\mathcal{X}_3| = O(n^4)$ holds by using the results in (i)–(ii). Assume $\lambda(G) > 0$. For the above two crossing subsets $X_1, X_2 \in \mathcal{X}_4$, we have

$$\begin{aligned}
opt(G,4) &\leq cost(Y_1; Y_2; Y_3; Y_4) \\
&= f(X_1) + f(X_2) - cost(Y_1; Y_3) - cost(Y_2; Y_4) \\
&\leq (1/2)opt(G,4) + (1/2)opt(G,4) - cost(Y_1; Y_3) - cost(Y_2; Y_4) \\
&= opt(G,4) - cost(Y_1; Y_3) - cost(Y_2; Y_4),
\end{aligned}$$

implying that $f(X_1) = f(X_2) = (1/2)opt(G,4)$ and $cost(Y_1; Y_3) = cost(Y_2; Y_4) = 0$. From this, the number of subsets X with $f(X) < (1/2)opt(G,4)$ is $O(n)$. Let $\mathcal{X}'_4 = \{X \in \mathcal{X}_4 \mid f(X) = (1/2)opt(G,4)\}$. If $\lambda(G) \geq (1/4)opt(G,4)$, then $f(X) = (1/2)opt(G,4) \leq 2\lambda(G)$ holds for every subset X with $f(X) = (1/2)opt(G,4)$, implying that $|\mathcal{X}'_4| = O(n^{2\alpha}) = O(n^4)$ holds for $\alpha = 2$. Assume $\lambda(G) < (1/4)opt(G,4)$. Let Z be a subset of V with $f(Z) = \lambda(G)$. Consider two crossing subsets $X_1, X_2 \in \mathcal{X}'_4$ with $Z \cap X_1 \neq \emptyset \neq Z \cap X_2$. As observed above, we have $f(X_1) = f(X_2) = (1/2)opt(G,4)$ and $cost(Y_1; Y_3) = cost(Y_2; Y_4) = 0$, indicating that one of the 3-way cuts $F_1 = (Y_1; Y_2; Y_3 \cup Y_4)$, $F_2 = (Y_1; Y_2 \cup Y_3; Y_4)$, $F_3 = (Y_1 \cup Y_2; Y_3; Y_4)$, and $F_4 = (Y_1 \cup Y_4; Y_2; Y_3)$ has cost at most $(3/4)opt(G,4)$. Such a 3-way cut F_i and $(Z; V-X)$ have $cost(F_i) + cost(Z; V-X) \leq (3/4)opt(G,4) + \lambda(G) < (3/4)opt(G,4) + (1/4)opt(G,4) = opt(G,4)$, and this means that $F_i \cup (Z; V-X)$ remains a 3-way cut, i.e., $X_1 \cap X_2 = Z$ (otherwise $F_i \cup (Z; V-X)$ would be a 4-way cut with cost less than $opt(G,4)$). Hence we have a property that, for two crossing subsets $X, X' \in \mathcal{X}'_4$ with $Z \cap X \neq \emptyset \neq Z \cap X'$, $X - Z$ and $X' - Z$ are disjoint, and we see that there are $O(n)$ subsets $X \in \mathcal{X}'_4$ with $Z \cap X \neq \emptyset$. Since there are $O(n^2)$ subsets Z with $f(Z) = \lambda(G)$, we have $|\mathcal{X}'_4| = O(n^2 \cdot n) = O(n^3)$. This completes the proof of the lemma. ∎

It is known that the h minimum 2-way cuts can be enumerated in $O(hnF(n,m))$ time [10, 11, 36]. Thus, for $k = 2, 3, 4$ (resp., $k \geq 5$), all subsets X with $f(X) \leq$

$(2/k)\mathit{opt}(G,k)$ (resp., $f(X) < (1/2)\mathit{opt}(G,k)$) can be obtained in $O(n^3F(n,m))$ time for $k=2$, in $O(n^5F(n,m))$ time for $k=3,4$ (by Lemma 7.1), and in $O(n^{2k-5}F(n,m))$ time for $k\geq 5$ (by Theorem 5.1).

We are ready to describe our algorithm for enumerating all minimum k-way cuts.

Algorithm ALL_CUTS(G,k)

Input: A graph $G=(V,E)$ and an integer $k\in[1,|V|]$.

Output: The set $\mathcal{F}$ of all minimum k-way cuts in G.

1. **if** $\mathit{opt}(G,k)=0$ **then** Return $\mathcal{F}:=\emptyset$
2. **else** /* $\mathit{opt}(G,k)>0$ */
3.   **if** $k=2$ **then** Return $\{(X;V-X)\mid f(X)=\lambda(G)\}$
4.   **else if** $k\in\{3,4\}$ **then**
    Compute the set $\mathcal{X}_k$ of all subsets $X\subset V$ such that
    $f(X)\leq(2/k)\mathit{opt}(G,k)$ and $|V-X|\geq k-1$;
5     $\mathcal{F}:=\{(X;V-X)\cup F_2\mid X\in\mathcal{X}_k,\ F_2\in \mathrm{ALL\_CUTS}(G[V-X],k-1)\}$;
6.   Return $\mathcal{F}:=\mathcal{F}-\{F\in\mathcal{F}\mid \mathit{cost}(F)>\mathit{opt}(G,k)\}$ /* $\min_{F\in\mathcal{F}}\mathit{cost}(F)=\mathit{opt}(G,k)$ */
7.   **else** /* $k\geq 5$ */
8.   Compute the set $\mathcal{X}_k$ of all subsets $X\subset V$ such that $f(X)<(1/2)\mathit{opt}(G,k)$;
9. $p:=\lceil(k-\sqrt{k})/2\rceil-1$;
10. $\mathcal{F}:=\bigcup_{X\in\mathcal{X}_k:|X|\geq p,\ |V-X|\geq k-p}\{F_1\cup F_2\mid$
        $F_1\in\mathrm{ALL\_CUTS}(G[X],p),\ F_2\in\mathrm{ALL\_CUTS}(G[V-X],k-p)\}$;
11.   Return $\mathcal{F}:=\mathcal{F}-\{F\in\mathcal{F}\mid \mathit{cost}(F)>\mathit{opt}(G,k)\}$ /* $\min_{F\in\mathcal{F}}\mathit{cost}(F)=\mathit{opt}(G,k)$ */
12.   **end** /* if */
13.   **end** /* if */
14. **end**. /* if */

We have the same recursive formula on the number of instances generated during the execution of ALL_CUTS, where the difference from the analysis for the runtime of MULTIWAY is some of initial terms in $M(k)$. By checking a solution to the formula up to 1.3×10^6 by a computer program, we see that the total number of instances is $O(n^{4k/(1-1.71/\sqrt{k})-20})$. Since each subset $X\in\mathcal{X}_k$ is generated in $O(nF(n,m))=O(n^4)$ time, we establish the next result.

THEOREM 7.2. *For a graph $G=(V,E)$ with n vertices m edges and an integer $k\in[3,n]$,* ALL_CUTS(G,k) *finds all minimum k-way cuts in $O(n^{4k/(1-1.71/\sqrt{k})-19}F(n,m))$ time, where $F(n,m)$ denotes the time complexity for computing a maximum flow in a graph with n vertices and m edges.*

8. Concluding remarks.

In this paper, we have shown that, for general k, all minimum k-way cuts can be computed in $O(n^{4k/(1-1.71/\sqrt{k})-16})$ time. This is the first deterministic algorithm with time complexity whose exponent is $O(k)$ for a general graph. The new time bound improves the previous time bound $O(n^{k^2/2-3k/2+4}F(n,m))$ for deterministic algorithms but is still higher than the bound $O(n^{2(k-1)}\log^3 n)$ of a Monte Carlo algorithm due to Karger and Stein [23]. So, it is left open to derive a better upper bound on the number of subsets X with $f(X)<\mathit{opt}(G,k)/2$. The key property for our algorithm is Theorem 5.1. A similar property is found in the article by Goldschmidt and Hochbaum [8], but we have a simpler proof for this property under a less constrained setting, which allows us to apply Lemma 3.1 to generate instances with nearly halved k. However, based on Theorem 5.1, we can find *all* minimum k-way cuts, requiring a high time complexity. Another future work is to find a more restricted characterization for a family $\mathcal{X}$ of subsets X from which we can construct *at least one* minimum k-way cut.

Acknowledgments. The authors would like to thank the referees for their thorough reading of this article, which led to many useful suggestions for improving presentation, and the editor for the effort of coordinating.

REFERENCES

[1] C. J. Alpert and A. B. Kahng, *Recent directions in netlist partitioning: A survey*, Integration, The VLSI Journal, 19 (1995), pp. 1–81.

[2] D. Bertsimas, C. P. Teo, and R. Vohra, *Analysis of LP relaxations for multiway and multicut problems*, Networks, 34 (1999), pp. 102–114.

[3] C. J. Colbourn, *The Combinatorics of Network Reliability*, Oxford University Press, Oxford, UK, 1987.

[4] W. H. Cunningham, *Optimal attack and reinforcement of a network*, J. ACM, 32 (1985), pp. 549–561.

[5] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, *The complexity of multiterminal cuts*, SIAM J. Comput., 23 (1994), pp. 864–894.

[6] W. E. Donath, *Logic partitioning*, in Physical Design Automation of VLSI Systems, B. T. Preas and M. J. Lorenzetti, eds., Benjamin Cummings, Menlo Park, CA, 1988, pp. 65–86.

[7] A. V. Goldberg and R. E. Tarjan, *A new approach to the maximum-flow problem*, J. ACM, 35 (1998), pp. 921–940.

[8] O. Goldschmidt and D. S. Hochbaum, *Polynomial algorithm for the k-cut problem for fixed k*, Math. Oper. Res., 19 (1994), pp. 24–37.

[9] D. Gusfield, *Connectivity and edge-disjoint spanning trees*, Inform. Process. Lett., 16 (1983), pp. 87–89.

[10] H. W. Hamacher, J.-C. Picard, and M. Queyranne, *Ranking the cuts and cut-sets of a network*, in Algebraic and Combinatorial Methods in Operations Research, North–Holland Math. Stud. 95, North–Holland, Amsterdam, 1984, pp. 183–200.

[11] H. W. Hamacher, J.-C. Picard, and M. Queyranne, *On finding the K best cuts in a network*, Oper. Res. Lett., 2 (1984), pp. 303–305.

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

[13] D. Hartvigsen, *Minimum path basis*, J. Algorithms, 15 (1993), pp. 125–142.

[14] D. Hartvigsen, *The planar multiterminal cut problem*, Discrete Appl. Math., 85 (1998), pp. 203–222.

[15] X. He, *An improved algorithm for the planar 3-cut problem*, J. Algorithms, 12 (1991), pp. 23–37.

[16] S. Kapoor, *On minimum 3-cuts and approximating k-cuts using cut trees*, in Integer Programming and Combinatorial Optimization, Lecture Notes in Comput. Sci. 1084, Springer-Verlag, Berlin, 1996, pp. 132–146.

[17] Y. Kamidoi, S. Wakabayashi, and N. Yoshida, *Faster algorithms for finding a minimum k-way cut in a weighted graph*, in Proceedings of the IEEE International Symposium on Circuits and Systems, IEEE Circuits and Systems Society, Piscataway, NJ, 1997, pp. 1009–1012.

[18] Y. Kamidoi, S. Wakabayashi, and N. Yoshida, *A divide-and-conquer approach to the minimum k-way cut problem*, Algorithmica, 32 (2002), pp. 262–276.

[19] D. R. Karger, *Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm*, in Proceedings of the ACM–SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 1993, pp. 21–30.

[20] D. R. Karger, *Minimum cuts in near-linear time*, in Proceedings of the 28th ACM Symposium on Theory of Computing, ACM, New York, 1996, pp. 56–63.

[21] D. R. Karger, P. Klein, C. Stein, M. Thorup, and N. Young, *Rounding algorithms for a geometric embedding of minimum multiway cut*, in Proceedings of the 31st ACM Symposium on Theory of Computing, ACM, New York, 1999, pp. 668–678.

[22] D. R. Karger and C. Stein, *An $\tilde{O}(n^2)$ algorithm for minimum cuts*, in Proceedings of the 25th ACM Symposium on Theory of Computing, ACM, New York, 1993, pp. 757–765.

[23] D. R. Karger and C. Stein, *A new approach to the minimum cut problems*, J. ACM, 43 (1996), pp. 601–640.

[24] A. V. Karzanov, *Determining the maximal flow in a network by the method of preflows*, Soviet Math. Dokl., 15 (1974), pp. 434–437.

[25] C. H. Lee, M. Kim, and C. I. Park, *An efficient k-way graph partitioning algorithm for task allocation in parallel computing systems*, in Proceedings of the IEEE International Conference on Computer-Aided Design, IEEE Computer Society, Los Alamitos, CA, 1990, pp. 748–751.

[26] T. Lengaur, *Combinatorial Algorithms for Integrated Circuit Layout*, Wiley, New York, 1990.

[27] M. S. Levine, *Faster randomized algorithms for computing minimum* $\{3, 4, 5, 6\}$*-way cuts*, in Proceedings of the ACM–SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2000, pp. 735–742.

[28] H. Nagamochi and T. Ibaraki, *Computing the edge-connectivity in multigraphs and capacitated graphs*, SIAM J. Discrete Math., 5 (1992), pp. 54–66.

[29] H. Nagamochi and T. Ibaraki, *A fast algorithm for computing minimum* 3*-way and* 4*-way cuts*, in Integer Programming and Combinatorial Optimization, Lecture Notes in Comput. Sci. 1610, Springer-Verlag, 1999, pp. 377–390.

[30] H. Nagamochi and T. Ibaraki, *A fast algorithm for computing minimum* 3*-way and* 4*-way cuts*, Math. Program., 88 (2000), pp. 507–520.

[31] H. Nagamochi, S. Katayama, and T. Ibaraki, *A faster algorithm for computing minimum* 5*-way and* 6*-way cuts in graphs*, J. Comb. Optim., 4 (2000), pp. 151–169.

[32] J. Naor and Y. Rabani, *Tree packing and approximating k-cuts*, in Proceedings of the ACM–SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2001, pp. 26–27.

[33] H. Saran and V. V. Vazirani, *Finding k cuts within twice the optimal*, SIAM J. Comput., 24 (1995), pp. 101–108.

[34] H. S. Stone, *Multiprocessor scheduling with the aid of network flow algorithms*, IEEE Trans. Software Engrg., 3 (1977), pp. 85–93.

[35] P. Tittmann, *Partitions and network reliability*, Discrete Appl. Math., 95 (1999) pp. 445–453.

[36] V. Vazirani and M. Yannakakis, *Suboptimal cuts: Their enumeration, weight, and number*, in Algebraic and Logic Programming, Lecture Notes in Comput. Sci. 632, Springer-Verlag, Berlin, 1992, pp. 366–377.

[37] W. C. Yeh, *A simple algorithm for the planar multiway put problem*, J. Algorithms, 39 (2001), pp. 68–77.

[38] L. Zhao, H. Nagamochi, and T. Ibaraki, *Approximating the minimum k-way cut in a graph via minimum* 3*-way cuts*, J. Comb. Optim., 5 (2001), pp. 397–410.

[39] L. Zhao, H. Nagamochi, and T. Ibaraki, *On generalized greedy splitting algorithms for multiway partition problems*, Discrete Appl. Math., 143 (2004), pp. 130–143.

[40] L. Zhao, H. Nagamochi, and T. Ibaraki, *Greedy splitting algorithms for approximating multiway partition problems*, Math. Program., 102 (2005), pp. 167–183.