Algorithmica (2002) 32: 262–276
DOI: 10.1007/s00453-001-0070-2

Algorithmica

A Divide-and-Conquer Approach to the Minimum *k*-Way Cut Problem

Y. Kamidoi,[1] S. Wakabayashi,[2] and N. Yoshida[1]

Abstract. This paper presents algorithms for computing a minimum 3-way cut and a minimum 4-way cut of an undirected weighted graph G. Let $G = (V, E)$ be an undirected graph with n vertices, m edges, and positive edge weights. Goldschmidt and Hochbaum presented an algorithm for the minimum k-way cut problem with fixed k, that requires $O(n^4)$ and $O(n^6)$ maximum flow computations, respectively, to compute a minimum 3-way cut and a minimum 4-way cut of G. In this paper we first show some properties on minimum 3-way cuts and minimum 4-way cuts, which indicate a recursive structure of the minimum k-way cut problem when $k = 3$ and 4. Then, based on those properties, we give divide-and-conquer algorithms for computing a minimum 3-way cut and a minimum 4-way cut of G, which require $O(n^3)$ and $O(n^4)$ maximum flow computations, respectively.

Key Words. Deterministic algorithm, Minimum k-way cut, Maximum flow computation, Undirected weighted graph, Recursive structure.

1. Introduction. Computing a minimum cut of a graph is one of the important problems in graph theory [4]. Let $G = (V, E)$ be an undirected graph with nonnegative edge weights. Given $k(\geq 2)$ disjoint nonempty subsets, $S_1, S_2, \ldots, S_k$, of V, an edge set $C \subseteq E$ is an $(S_1, S_2, \ldots, S_k)$-*terminal cut* of G if $G' = (V, E - C)$ has no paths from any $s \in S_i$ to any $t \in S_j$ if $i \neq j$. An edge set $C \subseteq E$ is a *k-way cut* of G if there are k disjoint vertex subsets, $Y_1, Y_2, \ldots, Y_{(k-1)}$ and Y_k, such that C is a $(Y_1, Y_2, \ldots, Y_k)$-terminal cut of G. The cost of a cut C is defined as the total of the edge costs in C. A k-way cut C is called *minimum* if it has the smallest cut cost among any k-way cuts of G. This paper discusses the problem of finding a minimum 3-way cut and a minimum 4-way cut of an undirected graph G.

Dahlhaus et al. [3] showed that the k-terminal cut problem is NP-hard for arbitrary k and even for $k = 3$. They also proposed a minimum k-terminal cut algorithm for a planar undirected graph. Gomory and Hu [6] showed that $O(n)$ executions of a minimum 2-terminal cut algorithm is enough to compute a minimum 2-way cut of an undirected graph. Goldschmidt and Hochbaum [5] showed a polynomial time algorithm for computing a minimum k-way cut for fixed k. This result showed that the k-way cut problem is easier than the k-terminal cut problem for an undirected graph. In their algorithm the minimum 2-terminal cut algorithm is repeatedly applied. The algorithm

[1] Faculty of Information Sciences, Hiroshima City University, 3-4-1 Ozuka-higashi, Asaminami-ku, Hiroshima 731-3194, Japan. yoko@ce.hiroshima-cu.ac.jp.
[2] Graduate School of Engineering, Hiroshima University, 4-1 Kagamiyama 1 chome, Higashi-Hiroshima 739-8597, Japan. wakaba@ecs.hiroshima-u.ac.jp.

Received May 1998; revised December 1999. Communicated by C. H. Papadimitriou.
Online publication October 19, 2001.

for the minimum k-way cut problem with fixed k applies $O(n^{k^2/2-3k/2+4})$ maximum flow computations for fixed k. This method finds a set of 2-way cuts whose cardinality is $O(n^{2k-3})$ as a set of 2-way cuts of candidates for constructing a minimum k-way cut. Nagamochi and Ibaraki [11] showed that the minimum 2-way cut problem on a graph with n vertices and m edges can be solved in $O(nm)$ computation time. Saran and Vazirani [14] proposed two approximation algorithms for the minimum k-way cut problem. One algorithm requires $n - 1$ maximum flow computations for finding a set of twice-optimal k-way cuts, one for each value of k between 2 and n. Kapoor [9] gave an algorithm for finding a minimum 3-way cut, which requires $O(n^3)$ maximum flow computations. Kapoor also gave an approximation technique for the multiway cut problem, and showed an algorithm for the minimum k-way cut problem, which requires $O(kn(m + n \log n))$ steps and gave an approximation of $2(1 - 1/k)$. Hochbaum and Shimoys [7] gave an $O(n^2)$ algorithm for finding a minimum 3-way cut of an unweighted planar graph. Recently, Burlet and Goldschmidt [2] presented an $O(mn^3)$ time algorithm for finding a minimum 3-way cut in an undirected weighted graph. They used the algorithm by Nagamochi et al. [13] that enumerates all cuts of weights less than l times the cost of a minimum 2-way cut without using the maximum flow algorithm, where $l \geq 1$.

All the algorithms shown above are ordinary deterministic algorithms, and thus they can always find optimal solutions or approximation solutions. On the other hand, Karger and Stein [10] proposed a randomized Monte Calro algorithm which finds a minimum 2-way cut with high probability in $O(n^2 \log^3 n)$. They also gave a randomized Monte Calro algorithm for the minimum k-way cut problem, which solves the problem in $O(n^{2(k-1)} \log^3 n)$ time.

Most of the previous algorithms for the minimum k-way cut problems adopted the same approach. Let C be a minimum k-way cut in $G = (V, E)$ separating V into k components $V_1, V_2, \ldots, V_k$, and let $\omega(V_i)$ be the weight of the set of edges that separates V_i from the rest of graph. Assume that $\omega(V_1) \leq \omega(V_i)$, $i \neq 1$. In most previous algorithms, a collection of vertex sets that necessarily includes the V_1 set of a minimum k-way cut is enumerated, and, for each vertex set, the minimum $(k-1)$-problem in both V_1 and $V - V_1$ is recursively solved.

Contrary to this common approach, in this paper we present a new approach, called a divide-and-conquer method. First, we show several properties on minimum 3-way cuts and minimum 4-way cuts, which indicate a recursive structure of the minimum k-way cut problem when $k = 3$ and 4. Then, based on the recursive properties, we present a divide-and-conquer strategy for the minimum 3-way and 4-way cut problems, and propose two polynomial time algorithms, each of which computes a minimum 3-way cut and a minimum 4-way cut of G, respectively. These algorithms require $O(n^3)$ and $O(n^4)$ maximum flow computations, respectively. This computation time of the proposed algorithms compares very favorably with the computation time of the previous methods, in particular, for $k = 4$.

The rest of our paper is organized as follows. The terminology is introduced in Section 2. In Section 3 we show the recursive properties on the minimum 3-way and 4-way cut problems. Section 4 presents a divide-and-conquer approach to the problems, and optimal algorithms for the problems are presented in Section 5. Their time complexity is also analyzed. Finally, in Section 6, we offer some conclusions.

2. Preliminaries. In the following we give some definitions and terminologies.

DEFINITION 1. Let $G = (V, E)$ be an undirected graph consisting of a vertex set V and an edge set E with an edge cost function $c\colon E \to R$, where R is the set of positive real numbers. Each edge $e \in E$ is incident to the elements of an unordered pair of vertices $u, v \in V$, called the end vertices of e, and the edge e is then denoted by $e = (u, v)$. For each edge $e \in E$, $c(e)$ represents the cost of e. Let $n = |V|$ be the number of vertices and let $m = |E|$ be the number of edges.

DEFINITION 2. Let $G = (V, E)$ be an undirected graph. For a nonempty subset $E' \subseteq E$, we denote the cost of E' by $c(E')$, which is defined as $\sum_{e \in E'} c(e)$.

DEFINITION 3. Let $G = (V, E)$ be an undirected graph and let $k (\geq 2)$ be an integer. Let $X_1, X_2, \ldots, X_k$ be k mutually disjoint nonempty subsets of V such that $V = X_1 \cup X_2 \cup \cdots \cup X_k$. Let $(X_1; X_2; \ldots; X_k)$ be the set of all those edges of G having one end vertex in X_i and the other in X_j, where $i \neq j$ and $1 \leq i, j \leq k$. The set $(X_1; X_2; \ldots; X_k)$ is called a k-way cut of G.

DEFINITION 4. Given an undirected graph $G = (V, E)$ and k mutually disjoint nonempty subsets of V, called terminal sets, denote $T_1, T_2, \ldots, T_k$ such that $T_1 \cup T_2 \cup \cdots \cup T_k \subseteq V$, an edge set $C \subseteq E$ is called a $(T_1, T_2, \ldots, T_k)$-terminal cut of G if the removal of C from E disconnects each terminal set from all the others. If every terminal set is a singleton set, i.e., $T_i = \{t_i\}$, then we simply represent a $(T_1, T_2, \ldots, T_k)$-terminal cut as a $(t_1, t_2, \ldots, t_k)$-terminal cut. If there is no edge set $C' \subset C$ such that C' is a $(T_1, T_2, \ldots, T_k)$-terminal cut of G, we call the cut C minimal. For a given $(T_1, T_2, \ldots, T_k)$-terminal cut C of G, if there is no $(T_1, T_2, \ldots, T_k)$-terminal cut C' of G such that $c(C') < c(C)$, we call the cut C minimum.

DEFINITION 5. For a given k-way cut C, C is minimum if $c(C) \leq c(C')$ for any k-way cut C'.

Given an undirected graph $G = (V, E)$ and k mutually disjoint nonempty subsets of V, we call the problem of finding a minimum $(T_1, T_2, \ldots, T_k)$-terminal cut of G the *minimum k-terminal cut problem*. Given an undirected graph $G = (V, E)$ and an integer $k\,(\geq 2)$, we call the problem of finding a minimum k-way cut of G the *minimum k-way cut problem*. From definitions, any minimal $(T_1, T_2, \ldots, T_k)$-terminal cut C can be represented as a k-way cut $(V_1; V_2; \ldots; V_k)$ where $T_i \subseteq V_i$, $1 \leq i \leq k$, and $V_1 \cup V_2 \cup \cdots \cup V_k = V$.

DEFINITION 6. Let $G = (V, E)$ be an undirected graph. Given a nonempty vertex subset X, let $G(X) = (X, E_X)$ be an induced subgraph of G by X with the edge cost function c_X such that for any edge $e \in E_X$, $c_X(e) = c(e)$.

Let X be a subset of vertices of $G = (V, E)$. $\bar{X}$ is the complement of X, i.e., $\bar{X} = V - X$.

DEFINITION 7. For an undirected graph $G = (V, E)$, let $C = (X; \bar{X})$ and $D = (Y; \bar{Y})$ be 2-way cuts of G. C is said to be intersected with D if the following four equations hold:

$$\bar{X} \cap \bar{Y} \neq \emptyset, \qquad \bar{X} \cap Y \neq \emptyset, \qquad X \cap \bar{Y} \neq \emptyset, \qquad X \cap Y \neq \emptyset.$$

Itcan be shown that if there exists a minimum k-terminal cut algorithm for G, we can solve the minimum k-way cut problem in polynomial time by applying it in $O(n^{k-1})$ times. For example, if $k = 2$, the minimum k-terminal problem becomes the famous minimum (s, t)-terminal cut problem, which can be solved in polynomial time based on the Ford–Fulkerson min-cut max-flow theorem [1]. Thus, the minimum 2-way cut problem can be solved by applying the min-cut max-flow algorithm $O(n)$ times. Dahlhaus et al. showed, however, that for even a fixed constant $k(\geq 3)$, the minimum k-terminal cut problem for a general graph is NP-hard [3]. So, it is hopeless to devise a minimum k-way cut algorithm based on a minimum k-terminal cut problem. For the general minimum k-way cut problem, we should adopt another approach. In this paper we present a divide-and-conquer approach to the minimum k-way cut problem when $k = 3$ and $k = 4$, and propose polynomial time algorithms.

3. Properties.

In this section we show several properties on minimum 3-way cuts and minimum 4-way cuts of G. In the next section these properties are used to derive a divide-and-conquer strategy to solve the minimum 3-way and 4-way cut problems. For any k-way cut $C = (S_1; S_2; \ldots; S_k)$, we denote the cost of C, $c(C) = \sum_{e \in C} c(e)$, by $c(S_1; S_2; \ldots; S_k)$.

3.1. *Properties on* 3-*Way Cuts*.

Given an undirected graph $G = (V, E)$, let $c_{2\min}$ and $c_{3\min}$ be the costs of a minimum 2-way cut and a minimum 3-way cut of G, respectively. Then the following lemma holds.

LEMMA 1. *Let $G = (V, E)$ be an undirected graph. For any minimum 3-way cut $(R; S; T)$ of G, the following holds*:

$$c_{2\min} \leq \min\{c(R; \bar{R}), c(S; \bar{S}), c(T; \bar{T})\} \leq \tfrac{2}{3} c_{3\min}.$$

Assume that there is a minimum 3-way cut $(R; S; T)$ of G such that $c(R; \bar{R}) = \min\{c(R; \bar{R}), c(S; \bar{S}), c(T; \bar{T})\}$. Let $(X; \bar{X})$ be a 2-way cut of G. Then, depending on the relation between $(R; \bar{R})$ and $(X; \bar{X})$, the following lemmas hold.

LEMMA 2. *Given a graph $G = (V, E)$ and a 2-way cut $(X; \bar{X})$ of G such that $c(X; \bar{X}) \leq \frac{2}{3} c_{3\min}$, if there is a minimum 3-way cut $(R; S; T)$ of G such that $c(R; \bar{R}) \leq \frac{2}{3} c_{3\min}$ and $(R; \bar{R})$ is intersected with $(X; \bar{X})$, then at least one of $(X; \bar{X} \cap R; \bar{X} \cap \bar{R})$ or $(\bar{X}; X \cap R; X \cap \bar{R})$ is a minimum 3-way cut of G.*

PROOF. From the assumption,

$$\min\{c((R \cap X); (\bar{R} \cap X)), c((R \cap \bar{X}); (\bar{R} \cap \bar{X}))\} \leq \tfrac{1}{2} c(R; \bar{R}) \leq \tfrac{1}{3} c_{3\min}.$$

If $c((R \cap X); (\bar{R} \cap X)) \leq c((R \cap \bar{X}); (\bar{R} \cap \bar{X}))$, then

$$\begin{aligned} c(\bar{X}; (R \cap X); (\bar{R} \cap X)) &= c(X; \bar{X}) + c((R \cap X); (\bar{R} \cap X)) \\ &\leq \tfrac{2}{3}c_{3\min} + \tfrac{1}{3}c_{3\min} \\ &= c_{3\min}. \end{aligned}$$

Thus, $(\bar{X}; (R \cap X); (\bar{R} \cap X))$ is a minimum 3-way cut of G.

If $c((R \cap \bar{X}); (\bar{R} \cap \bar{X})) \leq c((R \cap X); (\bar{R} \cap X))$, then we have a similar discussion to show that a 3-way cut $(X; (R \cap \bar{X}); (\bar{R} \cap \bar{X}))$ is a minimum 3-way cut of G. □

LEMMA 3. *Given a graph $G = (V, E)$ and a 2-way cut $(X; \bar{X})$ of G, if there is a minimum 3-way cut of G, denoted $(R; S; T)$, such that $c(X; \bar{X}) \leq c(R; \bar{R})$, $R \subseteq X$, $\bar{X} \cap S \neq \emptyset$, and $\bar{X} \cap T \neq \emptyset$, then $(X; Y; Z)$ is a minimum 3-way cut of G, where $(Y; Z)$ is a minimum 2-way cut of $G(\bar{X})$.*

PROOF.

$$\begin{aligned} c(X; Y; Z) &= c(X; \bar{X}) + c(Y; Z) \\ &\leq c(X; \bar{X}) + c((\bar{X} \cap S); (\bar{X} \cap T)) \\ &\leq c(R; \bar{R}) + c(S; T) \\ &= c(R; S; T). \end{aligned}$$

Thus, the lemma holds. □

3.2. *Properties on 4-Way Cuts*. Let $c_{4\min}$ be the cost of a minimum 4-way cut of G. Then the following lemma holds.

LEMMA 4. *Let $G = (V, E)$ be an undirected graph. For any minimum 4-way cut $(R; S; T; U)$ of G, the following holds*:

$$c_{2\min} \leq \min\{c(R; \bar{R}), c(S; \bar{S}), c(T; \bar{T}), c(U; \bar{U})\} \leq \tfrac{1}{2}c_{4\min}.$$

Assumethat there is a minimum 4-way cut $(R; S; T; U)$ of G such that $c(R; \bar{R}) = \min\{c(R; \bar{R}), c(S; \bar{S}), c(T; \bar{T}), c(U; \bar{U})\}$. Let $(X; \bar{X})$ be a 2-way cut of G. Then, depending on the relation between $(R; \bar{R})$ and $(X; \bar{X})$, Lemmas 5–8 hold.

LEMMA 5. *Given a graph $G = (V, E)$ and a 2-way cut $(X; \bar{X})$ of G such that $c(X; \bar{X}) \leq \frac{1}{2}c_{4\min}$, if there is a minimum 4-way cut $(R; S; T; U)$ of G such that $c(R; \bar{R}) \leq \frac{1}{2}c_{4\min}$ and $(R; \bar{R})$ is intersected with $(X; \bar{X})$, then $(X \cap R; X \cap \bar{R}; \bar{X} \cap R; \bar{X} \cap \bar{R})$ is a minimum 4-way cut of G.*

PROOF. From the assumption, $(X \cap R; X \cap \bar{R}; \bar{X} \cap R; \bar{X} \cap \bar{R})$ is indeed a 4-way cut of G. We have

$$\begin{aligned} c(X \cap R; X \cap \bar{R}; \bar{X} \cap R; \bar{X} \cap \bar{R}) &\leq c(X; \bar{X}) + c(R; \bar{R}) \\ &\leq \tfrac{1}{2}c_{4\min} + \tfrac{1}{2}c_{4\min} \\ &= c_{4\min}. \end{aligned}$$

Thus, the lemma holds. □

LEMMA 6. *Given an undirected graph $G = (V, E)$ and a 2-way cut, $(X; \bar{X})$, of G, if there is a minimum 4-way cut, $(R; S; T; U)$, of G such that $R \subseteq X$, $c(X; \bar{X}) \le c(R; \bar{R})$, $\bar{X} \cap S \neq \emptyset$, $\bar{X} \cap T \neq \emptyset$, and $\bar{X} \cap U \neq \emptyset$, then there is a minimum 4-way cut, denoted $(X; Y; Z; W)$, such that $(Y; Z; W)$ is a minimum 3-way cut of $G(\bar{X})$.*

PROOF. Since $R \subseteq X$, we have $\bar{X} \subseteq S \cup T \cup U$. Thus,

$$\begin{aligned} c(X; Y; Z; W) &= c(X; \bar{X}) + c(Y; Z; W) \\ &\le c(X; \bar{X}) + c(\bar{X} \cap S; \bar{X} \cap T; \bar{X} \cap U) \\ &\le c(R; \bar{R}) + c(S; T; U) \\ &= c(R; S; T; U). \end{aligned}$$

Thus, the lemma holds. □

LEMMA 7. *Given an undirected graph $G = (V, E)$, let $(X; \bar{X})$ be a 2-way cut of G. If there is a minimum 4-way cut, denoted $(R; S; T; U)$, of G such that $X = R \cup S$, then $(R'; S'; T'; U')$ is also a minimum 4-way cut of G, where $(R'; S')$ and $(T'; U')$ are minimum 2-way cuts of $G(X)$ and $G(\bar{X})$, respectively.*

PROOF. From the assumption, we have $X = R \cup S$ and $\bar{X} = T \cup U$. For $G(X)$, we have

$$c(R'; S') \le c(R; S).$$

For $G(\bar{X})$, we have

$$c(T'; U') \le c(T; U).$$

Then

$$\begin{aligned} c(R'; S'; T'; U') &= c(X; \bar{X}) + c(R'; S') + c(T'; U') \\ &\le c(R \cup S; T \cup U) + c(R; S) + c(T; U) \\ &= c(R; S; T; U). \end{aligned}$$

Thus, the lemma holds. □

LEMMA 8. *Given an undirected graph $G = (V, E)$, let $(X; \bar{X})$ be a 2-way cut of G. If there is a minimum 4-way cut, denoted $(R; S; T; U)$, of G such that $(X; \bar{X})$ is intersected with $(S; \bar{S})$ and $(T; \bar{T})$, $X \subset S \cup T$, and $c(X; \bar{X}) \le \min\{c(R; \bar{R}), c(S; \bar{S}), c(T; \bar{T}), c(U; \bar{U})\}$, then $(R'; S'; T'; U')$ is also a minimum 4-way cut of G, where $(R'; S')$ and $(T'; U')$ are minimum 2-way cuts of $G(X)$ and $G(\bar{X})$, respectively.*

PROOF. Without loss of generality, we assume that $c(S; \bar{S}) \le c(T; \bar{T})$. Since $(X; \bar{X})$ is intersected with $(S; \bar{S})$, $(S \cap X; S \cap \bar{X}; \bar{S} \cap X; \bar{S} \cap \bar{X})$ is a 4-way cut of G:

$$\begin{aligned} & c(S \cap X; S \cap \bar{X}; \bar{S} \cap X; \bar{S} \cap \bar{X}) \\ &\quad \le c(S; \bar{S}) + c(X; \bar{X}) \le \tfrac{1}{2}\{c(S; \bar{S}) + c(T; \bar{T})\} \\ &\qquad + \min\{c(R; \bar{R}), c(S; \bar{S}), c(T; \bar{T}), c(U; \bar{U})\} \\ &\quad \le \tfrac{1}{2}\{c(R; \bar{R}) + c(S; \bar{S}) + c(T; \bar{T}) + c(U; \bar{U})\} \\ &\quad = c_{4\,\min}. \end{aligned}$$

Thus, $(S \cap X; S \cap \bar{X}; \bar{S} \cap X; \bar{S} \cap \bar{X})$ is a minimum 4-way cut of G. Since $(S \cap X; \bar{S} \cap X)$ and $(S \cap \bar{X}; \bar{S} \cap \bar{X})$ are 2-way cuts of $G(X)$ and $G(\bar{X})$, respectively, we have

$$\begin{aligned} c(R'; S'; T'; U') &= c(X; \bar{X}) + c(R'; S') + c(T'; U') \\ &\leq c(X; \bar{X}) + c(S \cap X; \bar{S} \cap X) + c(S \cap \bar{X}; \bar{S} \cap \bar{X}) \\ &\leq c(S \cap X; S \cap \bar{X}; \bar{S} \cap X; \bar{S} \cap \bar{X}). \end{aligned}$$

Thus, the lemma holds. □

4. A Divide-and-Conquer Approach. In this section, first, we show a recursive structure of minimum 3-way cuts and minimum 4-way cuts of an undirected graph G. Then we present two main theorems, which will be a base to construct algorithms for computing a minimum 3-way cut and a minimum 4-way cut of G.

LEMMA 9. *Given an undirected graph $G = (V, E)$, let $(X; \bar{X})$ be a 2-way cut. Let $(Y; \bar{Y})$ and $(Z; \bar{Z})$ be minimum 2-way cuts of $G(\bar{X})$ and $G(X)$, respectively. If there is a minimum 3-way cut $(R; S; T)$ of G such that $c(X; \bar{X}) \leq \min\{c(R; \bar{R}), c(S; \bar{S}), c(T; \bar{T})\}$, then at least one of the following four properties holds*:

(i) *$(X; Y; \bar{Y})$ is a minimum 3-way cut of G.*
(ii) *$(\bar{X}; Z; \bar{Z})$ is a minimum 3-way cut of G.*
(iii) *There is a minimum 3-way cut, denoted $(R'; S'; T')$, such that $X \subset R'$.*
(iv) *There is a minimum 3-way cut, denoted $(R''; S''; T'')$, such that $\bar{X} \subset R''$.*

PROOF. Without loss of generality, we assume that $c(R; \bar{R}) = \min\{c(R; \bar{R}), c(S; \bar{S}), c(T; \bar{T})\}$. Consider the relation between $(X; \bar{X})$ and $(R; \bar{R})$. Then there are four cases. That is, (1) $(X; \bar{X})$ is intersected with $(R; \bar{R})$, (2) $X \subset R$, (3) $R \subseteq X$, and (4) $X \cap R = \emptyset$.

First, consider case (1). From Lemma 2, at least $(X; (\bar{X} \cap R); (\bar{X} \cap \bar{R}))$ or $(\bar{X}; (X \cap R); (X \cap \bar{R}))$ is a minimum 3-way cut of G. Consider the case that $(X; (\bar{X} \cap R); (\bar{X} \cap \bar{R}))$ is a minimum 3-way cut of G. In this case, $((\bar{X} \cap R); (\bar{X} \cap \bar{R}))$ is a 2-way cut of $G(\bar{X})$. Then, since $(Y; \bar{Y})$ is a minimum 2-way cut of $G(\bar{X})$, we have $c(Y; \bar{Y}) \leq c((\bar{X} \cap R); (\bar{X} \cap \bar{R}))$. Therefore, we have

$$c(X; Y; \bar{Y}) \leq c(X; (\bar{X} \cap R); (\bar{X} \cap \bar{R})) = c_{3\min}.$$

Thus, $(X; Y; \bar{Y})$ is a minimum 3-way cut of G. For the case that $(\bar{X}; (X \cap R); (X \cap \bar{R}))$ is a minimum 3-way cut of G, we have a similar discussion to show that $(\bar{X}; Z; \bar{Z})$ is a minimum 3-way cut of G. Consequently, for case (1), at least one of properties (i) or (ii) is satisfied.

Next, consider case (2). In this case it is clear that property (iii) is satisfied.

Next, consider case (3). This case is further classified into the following cases. That is, (3-1) $\bar{X} \cap S \neq \emptyset$ and $\bar{X} \cap T \neq \emptyset$, (3-2) there is a P such that $X = R \cup P$, $P \in \{S, T\}$, and (3-3) there is a Q such that $\bar{X} \subset Q$, $Q \in \{S, T\}$.

Consider case (3-1). From Lemma 3, we see that property (i) holds. Consider case (3-2). In this case it is clear that property (ii) holds. Consider case (3-3). In this case we see that property (iv) holds.

Finally, consider case (4). In this case we have $R \subseteq \bar{X}$. Let $X' = \bar{X}$. Then this is the same case as case (3). Thus, the lemma holds. □

LEMMA 10. *Given an undirected graph $G = (V, E)$, let $(X; \bar{X})$ be a 2-way cut. Let $(Y; \bar{Y})$ and $(Z; \bar{Z})$ be minimum 2-way cuts of $G(\bar{X})$ and $G(X)$, respectively. Let $(R; S; T)$ and $(R'; S'; T')$ be minimum 3-way cuts of $G(\bar{X})$ and $G(X)$, respectively. If there is a minimum 4-way cut $(A; B; C; D)$ of G such that $c(X; \bar{X}) \leq \min\{c(A; \bar{A}), c(B; \bar{B}), c(C; \bar{C}), c(D; \bar{D})\}$, then at least one of the following five properties holds*:

(i) *$(X; R; S; T)$ is a minimum 4-way cut of G.*
(ii) *$(\bar{X}; R'; S'; T')$ is a minimum 4-way cut of G.*
(iii) *$(Y; \bar{Y}; Z; \bar{Z})$ is a minimum 4-way cut of G.*
(iv) *There is a minimum 4-way cut, denoted $(A'; B'; C'; D')$, such that $X \subset A'$.*
(v) *There is a minimum 4-way cut, denoted $(A''; B''; C''; D'')$, such that $\bar{X} \subset A''$.*

PROOF. Without loss of generality, we assume that $c(A; \bar{A}) = \min\{c(A; \bar{A}), c(B; \bar{B}), c(C; \bar{C}), c(D; \bar{D})\}$. Consider the relation between $(X; \bar{X})$ and $(A; \bar{A})$. Then there are four cases. That is, (1) $(X; \bar{X})$ is intersected with $(A; \bar{A})$, (2) $X \subset A$, (3)$A \subseteq X$, and (4) $X \cap A = \emptyset$.

First, consider case (1). From Lemma 5, $(X \cap A; X \cap \bar{A}; \bar{X} \cap A; \bar{X} \cap \bar{A})$ is a minimum 4-way cut of G. Since $(X \cap A; X \cap \bar{A})$ and $(\bar{X} \cap A; \bar{X} \cap \bar{A})$ are 2-way cuts of $G(X)$ and $G(\bar{X})$, respectively, we have

$$\begin{aligned} c(Y; \bar{Y}; Z; \bar{Z}) &= c(X; \bar{X}) + c(Y; \bar{Y}) + c(Z; \bar{Z}) \\ &\leq c(X; \bar{X}) + c(\bar{X} \cap A; \bar{X} \cap \bar{A}) + c(X \cap A; X \cap \bar{A}) \\ &= c(X \cap A; X \cap \bar{A}; \bar{X} \cap A; \bar{X} \cap \bar{A}) \\ &= c_{4\,\min}. \end{aligned}$$

Thus, property (iii) holds.

Next, consider case (2). For this case, property (iv) holds.

Next, consider case (3). This case is further classified into the following four cases. That is, (3-1) $\bar{X} \cap B \neq \emptyset$, $\bar{X} \cap C \neq \emptyset$, and $\bar{X} \cap D \neq \emptyset$, (3-2) there are $L, M, N \in \{B, C, D\}$, $L \neq M$, $M \neq N$, $L \neq N$, such that $X \subseteq A \cup L \cup M$, $\bar{X} \subseteq M \cup N$, (3-3) there are $L, M, N \in \{B, C, D\}$, $L \neq M$, $M \neq N$, $L \neq N$, such that $X \subset A \cup L \cup M \cup N$, $\bar{X} \subset M \cup N$, $(X; \bar{X})$ is intersected with $(M; \bar{M})$ and $(N; \bar{N})$, and (3-4) there is $L \in \{B, C, D\}$ such that $\bar{X} \subset L$.

Consider case (3-1). From Lemma 6, property (i) holds. Consider case (3-2). If $X = A \cup L$ and $\bar{X} = M \cup N$, then from Lemma 7, property (iii) holds. Consider otherwise. Then we have $X \cap A \neq \emptyset$, $X \cap L \neq \emptyset$, and $X \cap M \neq \emptyset$. From the assumption, we have $c(X; \bar{X}) \leq c(N; \bar{N})$. Let $X' = \bar{X}$. Then we see from Lemma 6 that property (ii) holds. Next, consider case (3-3). Let $X' = \bar{X}$. Then, from Lemma 8, we see that property (iii) holds. Consider case (3-4). For this case, it is obvious that property (iv) holds.

Finally, consider case (4). Since $X \cap A = \emptyset$, we have $A \subseteq \bar{X}$. Let $X' = \bar{X}$. Then this is the same as case (3). Thus, the lemma holds. □

Lemmas 9 and 10 tell us that a minimum 3-way cut and a minimum 4-way cut can be computed recursively.

DEFINITION 8. Let u and v be distinct vertices of a graph $G = (V, E)$. We can construct a new graph G' by fusing the two vertices, namely, by replacing them by a single new vertex x such that every edge that was incident with u or v in G is now incident with x in G'. Given a subset X of V, let $Shrink(G, X)$ be a graph obtained by fusing all the vertices in X, and removing all the self-loop edges from the resulting graph.

From Lemmas 9 and 10, and the definition of $Shrink(G, X)$, we can show the following main theorems.

THEOREM 1. *Let $G = (V, E)$ be a graph, and let $(X; \bar{X})$ be a 2-way cut of G. Let $(Y; \bar{Y})$ and $(Z; \bar{Z})$ be minimum 2-way cuts of $G(\bar{X})$ and $G(X)$, respectively. Let $(R'; S'; T')$ be a minimum 3-way cut of $Shrink(G, X)$, and let $(R''; S''; T'')$ be a minimum 3-way cut of $Shrink(G, \bar{X})$. If there is a minimum 3-way cut $(R; S; T)$ of G such that $c(X; \bar{X}) \leq \min\{c(R; \bar{R}), c(S; \bar{S}), c(T; \bar{T})\}$, then at least one of the following 3-way cuts of G, $(X; Y; \bar{Y})$, $(\bar{X}; Z; \bar{Z})$, $(R'; S'; T')$, and $(R''; S''; T'')$, is a minimum 3-way cut of G.*

THEOREM 2. *Let $G = (V, E)$ be a graph, and let $(X; \bar{X})$ be a 2-way cut of G. Let $(Y; \bar{Y})$ and $(Z; \bar{Z})$ be minimum 2-way cuts of $G(\bar{X})$ and $G(X)$, respectively. Let $(R; S; T)$ and $(R'; S'; T')$ be minimum 3-way cuts of $G(\bar{X})$ and $G(X)$, respectively. Let $(A'; B'; C'; D')$ and $(A''; B''; C''; D'')$ be minimum 4-way cuts of $G(\bar{X})$ and $G(X)$, respectively. If there is a minimum 4-way cut $(A; B; C; D)$ of G such that $c(X; \bar{X}) \leq \min\{c(A; \bar{A}), c(B; \bar{B}), c(C; \bar{C}), c(D; \bar{D})\}$, then at least one of the following 4-way cuts of G, $(X; R; S; T)$, $(\bar{X}; R'; S'; T')$, $(Y; \bar{Y}; Z; \bar{X})$, $(A'; B'; C'; D')$ and $(A''; B''; C''; D'')$, is a minimum 4-way cut of G.*

5. Algorithms.

Based on Theorems 1 and 2, we can present simple divide-and-conquer algorithms for computing a minimum 3-way cut and a minimum 4-way cut of an undirected graph. From Theorems 1 and 2, we find a recursive structure of the minimum 3-way and 4-way cut problems. For example, consider the minimum 3-way cut problem. Then, given a graph $G = (V, E)$, we can find a minimum 3-way cut of G by computing some combinations of minimum 2-way cuts, or by computing minimum 3-way cuts of $Shrink(G, X)$ and $Shrink(G, \bar{X})$ for some 2-way cut $(X; \bar{X})$ of G. If both $Shrink(G, X)$ and $Shrink(G, \bar{X})$ are smaller than G in the number of vertices, then we see that the minimum 3-way cut problem can be solved in a divide-and-conquer manner. For some minimum 3-way and 4-way cuts, denoted $(R; S; T)$ and $(R'; S'; T'; U')$, let $c_{3-2\min} = \min\{c(R; \bar{R}), c(S; \bar{S}), c(T; \bar{T})\}$ and $c_{4-2\min} = \min\{c(R'; \bar{R}'), c(S'; \bar{S}'), c(T'; \bar{T}'), c(U'; \bar{U}')\}$. Then the problem we should consider is thus the following: How do we find a 2-way cut $(X; \bar{X})$ of G such that (i) $c(X; \bar{X}) \leq c_{3-2\min}$ or $c(X; \bar{X}) \leq c_{4-2\min}$, and (ii) $|X| \geq 2$ and $|\bar{X}| \geq 2$?

In the following we show a method for finding a 2-way cut satisfying the above condition.

LEMMA 11. *Given an undirected graph* $G = (V, E)$, *let* x_1, x_2, x_3, x_4 *be four distinct vertices in* V *such that an* $(\{x_1, x_2\}, \{x_3, x_4\})$*-terminal cut of* G *is minimum in its cost among all those* $(\{u, v\}, \{w, x\})$*-terminal cuts of* G *for any four distinct vertices,* u, v, w, x, *in* V. *We denote this* $(\{x_1, x_2\}, \{x_3, x_4\})$*-terminal cut by* $(X; \bar{X})$. *If there is a minimum* 3*-way cut* $(R; S; T)$ *of* G *satisfying* $|R|, |S|, |T| \geq 2$, *then* $c(X; \bar{X}) \leq c_{3-2\min}$, *where* $c_{3-2\min} = \min\{c(R; \bar{R}), c(S; \bar{S}), c(T; \bar{T})\}$. *If there is a minimum* 4*-way cut* $(R'; S'; T'; U')$ *of* G *satisfying* $|R'|, |S'|, |T'|, |U'| \geq 2$, *then* $c(X; \bar{X}) \leq c_{4-2\min}$, *where* $c_{4-2\min} = \min\{c(R'; \bar{R}'), c(S'; \bar{S}'), c(T'; \bar{T}'), c(U'; \bar{U}')\}$.

PROOF. Consider the case of finding a minimum 3-way cut. For the case of finding a minimum 4-way cut, we can prove the lemma by giving a similar discussion to that shown below. Without loss of generality, $c(R; \bar{R}) = c_{3-2\min}$. From the assumption of $(X; \bar{X})$, there are four distinct vertices x_1, x_2, x_3, x_4 such that $x_1, x_2 \in X$, $x_3, x_4 \in \bar{X}$, and $(X; \bar{X})$ is a minimum $(\{x_1, x_2\}, \{x_3, x_4\})$-terminal cut of G. Since $|R| \geq 2$ and $|\bar{R}| \geq 2$, we can choose two distinct vertices, say u and v, from R and two distinct vertices, say w and x, from $\bar{R}$. Let $(Y; \bar{Y})$ be a minimum $(\{u, v\}, \{w, x\})$-terminal cut of G. Then, from the assumption, it is always true that $c(X; \bar{X}) \leq c(Y; \bar{Y})$ holds for any $u, v \in R$, $u \neq v$, and $w, x \in \bar{R}$, $w \neq x$. Thus, the lemma holds. □

Based on Lemma 11, given a graph $G = (V, E)$, we present a procedure to find a 2-way cut, $(X; \bar{X})$, of G, which satisfies (i) $|X| \geq 2$ and $|\bar{X}| \geq 2$, and (ii) $c(X; \bar{X}) \leq c_{3-2\min}$ and $c(X; \bar{X}) \leq c_{4-2\min}$ for any 3-way, and 4-way cuts of G. A straightforward way to find $(X; \bar{X})$ would be as follows. We enumerate all the combinations of four distinct vertices of G, say u, v, w, and x, and for each set of vertices, we find a minimum $(\{u, v\}, \{w, x\})$-terminal cut of G. Among all the combinations of four vertices, we select one set of vertices, say $\{u', v', w', x'\}$, such that the cost of a minimum $(\{u', v'\}, \{w', x'\})$-terminal cut of G is minimum among all the other combinations of the four vertices. Then let $(X; \bar{X})$ be the minimum $(\{u', v'\}, \{w', x'\})$-terminal cut of G. Note that, for given distinct four vertices of G, finding a minimum $(\{u, v\}, \{w, x\})$-terminal cut of G is easy. First, we add two new vertices s and t to G, and then we add new edges (s, u), (s, v), (t, w), and (t, x). We define the cost of the new edges as ∞. Then we find a minimum (s, t)-terminal cut of G by applying a minimum 2-terminal cut algorithm.

The procedure shown above, however, would require $O(n^4)$ min-cut max-flow computations. In the following we show an efficient method to compute $(X; \bar{X})$, which requires $O(n^2)$ min-cut max-flow computations. First, we pay attention to the following fact.

FACT 1. *Given an undirected graph* $G = (V, E)$, *let* $(X; \bar{X})$ *be a* 2*-way cut of* G. *Let* $S = \{u, v, w, x\}$ *be four distinct vertices in* V. *Let* n_X *and* $n_{\bar{X}}$ *be the numbers of vertices in* S, *which are contained in* X *and* $\bar{X}$, *respectively. Then one of the following conditions holds*: (i) $n_X = n_{\bar{X}} = 2$. (ii) $\max\{n_X, n_{\bar{X}}\} = 3$ *and* $\min\{n_X, n_{\bar{X}}\} = 1$. (iii) $\max\{n_X, n_{\bar{X}}\} = 4$ *and* $\min\{n_X, n_{\bar{X}}\} = 0$.

This Fact gives the base of our algorithm for computing $(X; \bar{X})$. Assume that a fixed set of four distinct vertices, say $S_0 = \{u_0, v_0, w_0, x_0\}$, is given in advance. For any distinct four vertices of G, say $\{u, v, w, x\}$, consider a minimum $(\{u, v\}, \{w, x\})$-terminal cut of G, denoted $(Y; \bar{Y})$. Then, from Fact 1, one of the following cases holds:

Case 1. S_0 is partitioned into two subsets, say T and U, each of which consists of two elements, respectively, so that $(Y; \bar{Y})$ is a minimum (T, U)-terminal cut of G.

Case 2. S_0 is partitioned into two subsets, say T' and U', each of which consists of three and one elements, respectively, so that $(Y; \bar{Y})$ is a minimum $(T', U' \cup \{y\})$-terminal cut of G, where y is a vertex in G.

Case 3. S_0 is not partitioned so that $(Y; \bar{Y})$ is a minimum $(S_0, \{y, z\})$-terminal cut of G, where y and z are vertices in G.

From those results mentioned above, we present a procedure to find $(X; \bar{X})$, which satisfies the conditions given previously. We call this procedure, procedure *Divide*(G). A description of procedure *Divide*(G) is given below.

In the description, *MinCut*(T, U) is a function to find a minimum (T, U)-terminal cut of G, which invokes a min-cut max-flow algorithm in one computation.

Function *Divide*(G)

Input: an undirected graph $G = (V, E)$.

begin
choose four distinct vertices from V, and let them be $S = \{u, v, w, x\}$;
$\mathcal{C}_{\min} := \emptyset$; $c_{\min} := \infty$;
/* Case (1) */
$\mathcal{C}_1 := MinCut(\{u, v\}, \{w, x\})$;
if $c_{\min} \geq c(\mathcal{C}_1)$ **then** $\mathcal{C}_{\min} := \mathcal{C}_1$; $c_{\min} := c(\mathcal{C}_1)$;
$\mathcal{C}_2 := MinCut(\{u, w\}, \{v, x\})$;
if $c_{\min} \geq c(\mathcal{C}_2)$ **then** $\mathcal{C}_{\min} := \mathcal{C}_2$; $c_{\min} := c(\mathcal{C}_2)$;
$\mathcal{C}_3 := MinCut(\{u, x\}, \{v, w\})$;
if $c_{\min} \geq c(\mathcal{C}_3)$ **then** $\mathcal{C}_{\min} := \mathcal{C}_3$; $c_{\min} := c(\mathcal{C}_3)$;
/* Case (2) */
For each $y \in V - S$ **do**
$\mathcal{C}_1 := MinCut(\{u, v, w\}, \{x, y\})$;
if $c_{\min} \geq c(\mathcal{C}_1)$ **then** $\mathcal{C}_{\min} := \mathcal{C}_1$; $c_{\min} := c(\mathcal{C}_1)$;
$\mathcal{C}_2 := MinCut(\{u, v, x\}, \{w, y\})$;
if $c_{\min} \geq c(\mathcal{C}_2)$ **then** $\mathcal{C}_{\min} := \mathcal{C}_2$; $c_{\min} := c(\mathcal{C}_2)$;
$\mathcal{C}_3 := MinCut(\{u, w, x\}, \{v, y\})$;
if $c_{\min} \geq c(\mathcal{C}_3)$ **then** $\mathcal{C}_{\min} := \mathcal{C}_3$; $c_{\min} := c(\mathcal{C}_3)$;
$\mathcal{C}_4 := MinCut(\{v, w, x\}, \{u, y\})$;
if $c_{\min} \geq c(\mathcal{C}_4)$ **then** $\mathcal{C}_{\min} := \mathcal{C}_4$; $c_{\min} := c(\mathcal{C}_4)$;
/* Case (3) */
For each $y, z \in V - S$, $y \neq z$ **do**
$\mathcal{C}_1 := MinCut(\{u, v, w, x\}, \{y, z\})$;
if $c_{\min} \geq c(\mathcal{C}_1)$ **then** $\mathcal{C}_{\min} := \mathcal{C}_1$; $c_{\min} := c(\mathcal{C}_1)$;
/* End of function */
Return$(\mathcal{C}_{\min})$
end

For this procedure we can show the following theorem.

THEOREM 3. *The function Divide*(G) *finds a smallest cost cut in all minimum* $(\{u, v\}, \{w, x\})$*-terminal cuts for any four distinct vertices in* G, *by applying at most* $O(n^2)$ *min-cut max-flow computations*.

PROOF. The correctness of the function was derived from Fact 1, as we discussed previously. Since there are doubly nested loops on vertices in G, it is clear that the function *MinCut*, which executes the min-cut max-flow computation once, was invoked $O(n^2)$ times in total. □

5.1. *The* 3-*Way Cut Algorithm*. In this subsection we present an algorithm for computing a minimum 3-way cut of a given graph G. The proposed algorithm is based on Theorem 1. Note that there is a special case in which, for a given graph G, there is no minimum 3-way cut $(R; S; T)$ such that $|R| \geq 2$, $|S| \geq 2$, and $|T| \geq 2$. In such a case we cannot compute a minimum 3-way cut by applying the function *Divide*, and we should treat this case separately.

The following are functions which are used in the proposed algorithm. Note that, given a graph $G = (V, E)$, a minimum 2-way cut of G can be computed in $O(|V|)$ min-cut max-flow computations.

(i) MIN-ONE-TERM-3WAY(G) computes a smallest cost 3-way cut, whose structure is given as $(\{x\}; Y; \bar{Y})$, where x is a vertex in G and $(Y; \bar{Y})$ is a minimum 2-way cut of $G(V - \{x\})$. [$O(n^2)$ maximum flow computations.]

(ii) *Divide*(G) is a function to compute a 2-way cut $(X; \bar{X})$ of G such that (i) $|X| \geq 2$ and $|\bar{X}| \geq 2$, and (ii) $c(X; \bar{X}) \leq c_{3-2\,\min}$ and $c(X; \bar{X}) \leq c_{4-2\,\min}$ for any minimum 3-way and 4-way cuts of G. [$O(n^2)$ maximum flow computations.]

(iii) CONST-3WAY-CUT(X) constructs a 3-way cut C of G by combining $(X; \bar{X})$ and a minimum 2-way cut in $G(\bar{X})$. [$O(n)$ maximum flow computations.]

(iv) ENUMERATE-ALL-3CUTS(G) enumerates all 3-way cuts of G, and returns the one with the smallest cost.

Algorithm MIN-TRI-PARTITION(G)

Input: an undirected graph $G = (V, E)$.

begin
  $\mathcal{C}_0$ = MIN-ONE-TERM-3WAY(G); $\mathcal{C}_1$ = MIN-3WAY-CUT(G);
  return MIN$(\mathcal{C}_0, \mathcal{C}_1)$
end.

Recursive Procedure MIN-3WAY-CUT(G)

Input: an undirected graph $G = (V, E)$.

begin
if $|V| < 6$ **then** return ENUMERATE-ALL-3CUTS(G);
  else begin
    $(X; \bar{X}) \leftarrow$ *Divide*(G);
    $G_X \leftarrow$ *Shrink*(G, X); $G_{\bar{X}} \leftarrow$ *Shrink*$(G, \bar{X})$;

$\mathcal{C}_0 \leftarrow$ CONST-3WAY-CUT(X); $\mathcal{C}_1 \leftarrow$ CONST-3WAY-CUT($\bar{X}$);
$\mathcal{C}_2 \leftarrow$ MIN-3WAY-CUT(G_X); $\mathcal{C}_3 \leftarrow$ MIN-3WAY-CUT($G_{\bar{X}}$);
return MIN($\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$)
end
end

5.2. *The* 4-*Way Cut Algorithm*. In this subsection we present an algorithm for computing a minimum 4-way cut of a given graph G. The proposed algorithm is based on Theorem 2. Note that there is a special case in which, for given G, there is no minimum 4-way cut $(R; S; T; U)$ such that $|R| \geq 2$, $|S| \geq 2$, $|T| \geq 2$, and $|U| \geq 2$. In such a case we cannot compute a minimum 4-way cut by applying the function *Divide*, and we should treat this case separately.

The following are functions, which are used in the proposed algorithm:

(i) MIN-ONE-TERM-4WAY(G) computes a smallest cost 4-way cut $(\{x\}; Y; Z; W)$ in all 4-way cuts constructed by a minimum 3-way cut in $G(V - \{x\})$ and a 2-way cut $(\{x\}; V - \{x\})$, where $x \in V$. [$O(n^4)$ maximum flow computations.]
(ii) CONST-4WAY-CUT(X) constructs a 4-way cut $\mathcal{C}$ of G by using the combination of $(X; \bar{X})$ and a minimum 3-way cut in $G(\bar{X})$. [$O(n^3)$ maximum flow computations.]
(iii) OTHER-4WAY-CUT(X) constructs a 4-way cut $\mathcal{C}$ of G by using the combination of $(X; \bar{X})$, a minimum 2-way cut in $G(\bar{X})$ and a minimum two-way cut in $G(X)$. [$O(n)$ maximum flow computations.]
(iv) ENUMERATE-ALL-4CUTS(G) enumerates all 4-way cuts of G, and returns the one with the smallest cost.

Algorithm MIN-QUADRI-PARTITION(G)

Input: an undirected graph $G = (V, E)$.

begin
$\mathcal{C}_0$ = MIN-ONE-TERM-4WAY(G); $\mathcal{C}_1$ = MIN-4WAY-CUT(G);
return MIN($\mathcal{C}_0, \mathcal{C}_1$)
end.

Recursive Procedure MIN-4WAY-CUT(G)

Input: an undirected graph $G = (V, E)$.

begin
if $|V| < 6$ **then** return ENUMERATE-ALL-4CUTS(G);
else begin
$(X; \bar{X}) \leftarrow Divide(G)$;
$G_X \leftarrow Shrink(G, X)$; $G_{\bar{X}} \leftarrow Shrink(G, \bar{X})$;
$\mathcal{C}_0 \leftarrow$ CONST-4WAY-CUT(X); $\mathcal{C}_1 \leftarrow$ CONST-4WAY-CUT($\bar{X}$);
$\mathcal{C}_2 \leftarrow$ OTHER-4WAY-CUT(X);
$\mathcal{C}_3 \leftarrow$ MIN-4WAY-CUT(G_X); $\mathcal{C}_4 \leftarrow$ MIN-4WAY-CUT($G_{\bar{X}}$);
return MIN($\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$)
end
end

5.3. *Computation Time.* The correctness of the proposed algorithms can be easily shown from Theorems 1 and 2. For the time complexity of the algorithms, we show the following theorem.

THEOREM 4. *For an undirected graph $G = (V, E)$, the algorithm* MIN-TRI-PARTITION(G) *and the algorithm* MIN-QUADRI-PARTITION(G) *compute a minimum* 3-*way cut and a minimum* 4-*way cut by applying $O(n^3)$ and $O(n^4)$ maximum flow computations, respectively.*

PROOF. In the following we consider the computation time of the algorithm MIN-TRI-PARTITION(G). The computation time of the algorithm MIN-QUADRI-PARTITION (G) can be discussed similarly.

Given a graph $G = (V, E)$, let K be the total number of invocations of the procedure MIN-3WAY-CUT in the algorithm. Then, from the description of the algorithm, it is easy to show that the algorithm invokes the min-cut max-flow procedure $O(Kn^2)$ times. Thus, in the following, we derive an upper bound of K.

First, we define a rooted tree called *computation tree* $T = (N, A)$ as follows. Each vertex, v, of T has a weight, denoted $w(v)$. Each vertex in T corresponds to an invocation of MIN-3WAY-CUT in the algorithm. The root of T corresponds to the first invocation of MIN-3WAY-CUT, whose actual parameter is G itself. Assume that G' is an input graph of MIN-3WAY-CUT, and applying *Divide* to G', two new graphs, $G_X = Shrink(G', X)$ and $G_{\bar{X}} = Shrink(G', \bar{X})$, are produced. Then, in T, there are three vertices, u, v, and w, which correspond to G', G_X, and $G_{\bar{X}}$, respectively, and there are edges (u, v) and (u, w). The weights of u, v, and w are the number of vertices in G', G_X, and $G_{\bar{X}}$. For simplicity, we assume that in the algorithm, if a given graph has more than three vertices, then MIN-3WAY-CUT will be applied to continue the recursive calls of MIN-3WAY-CUT, although, in the actual algorithm, if a given graph has less than six vertices, the recursive calls will terminate. Then the weight of a vertex has the following properties: (i) Let r be the root of T. Then $w(r) = |V| = n$. (ii) For each internal vertex v, let u and w be its left and right sons, respectively. Then $w(v) \geq 4$, $w(u) \geq 3$, $w(w) \geq 3$, and $w(v) + 2 = w(u) + w(w)$. (iii) For each leaf v, $w(v) = 3$.

Now, it is clear that T is a full binary tree, i.e., a binary tree whose any internal vertex has left and right sons. Let $I(T)$ and $L(T)$ be the numbers of internal vertices and leafs, respectively. Then we can easily show that $L(T) = I(T) + 1$. Let *SUM* be the total of weights of all leafs. Then, from the properties of the weights of vertices, we can show that $SUM = w(r) + I(T) \times 2$. On the other hand, it is obvious that $SUM = L(T) \times 3$. Since $w(r) = n$, we have $n + I(T) \times 2 = L(T) \times 3$. Substituting the equation $L(T) = I(T) + 1$, we finally get $n + I(T) \times 2 = (I(T) + 1) \times 3$.

Thus, we have $n - 3 = I(T)$ and $L(T) = I(T) + 1 = n - 2$. Consequently, the total number of invocation of MIN-3WAY-CUT is $I(T) + L(T) = 2n - 5$. This shows that the algorithm MIN-TRI-PARTITION invokes the min-cut max-flow procedure $O(n^3)$ times. □

Note that there have been a number of min-cut max-flow algorithms [1]. The time complexity of finding a minimum (s, t)-terminal cut of a general undirected weighted graph G is bounded by $O(n^3)$.

6. Conclusion. We have presented divide-and-conquer algorithms for computing a minimum 3-way cut and a minimum 4-way cut of an undirected weighted graph, which require $O(n^3)$ and $O(n^4)$ maximum flow computations, respectively.

As future work, we will consider an extension of the proposed algorithms for the minimum k-way cut problem for $k \geq 5$. Recently, after submitting the manuscript of this paper to this journal, Nagamochi and Ibaraki [12] proposed an algorithm for the minimum k-way cut algorithm for $k = 3, 4$, whose computation time is $O(n^{k-2}(nF(n, m) + C_2(n, m) + n^2))$, where $F(n, m)$ and $C_2(n, m)$ denote respectively the time complexity required to solve the maximum flow problem and the minimum 2-way cut problem in G, indicating that this algorithm runs faster than ours. This method uses the algorithm [15] that finds all the 2-way cuts in G in the order of nondecreasing weights. Note that a preliminary version of this paper first appeared in [8]. Thus, the reduction of computation time of the proposed algorithm is another of our concerns for future work.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin: *Network Flows*: *Theory, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[2] M. Burlet and O. Goldschmidt: A new and improved algorithm for the 3-cut problem, *Oper. Res. Lett.*, 21: 225–227, 1997.

[3] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis: The complexity of multiterminal cuts, *SIAM J. Comput.*, 23(4): 864–894, 1994.

[4] S. Even: *Graph Algorithms*, Computer Science Press, Rockville, MD, 1979.

[5] O. Goldschmidt and D. S. Hochbaum: Polynomial algorithm for the k-cut problem for fixed k, *Math. Oper. Res.*, 19(1): 24–37, 1994.

[6] R. Gomory and T. C. Hu: Multi-terminal network flows, *J. SIAM*, 9: 551–570, 1961.

[7] D. Hochbaum and D. Shimoys: An $O(V^2)$ algorithm for the planar 3-cut problem, *SIAM J. Discrete Math.*, 6(4): 707–712, 1985.

[8] Y. Kamidoi, S. Wakabayashi, and N. Yoshida: Faster algorithms for finding a minimum k-way cut in a weighted graph, *Proc.* 1997 *IEEE Internat. Symp. on Circuits and Systems*, Vol. 2, pp. 1009–1012, 1997.

[9] S. Kapoor: *On Minimum* 3*-Cuts and Approximating k-Cuts Using Cut Trees*, Lecture Notes in Computer Science 1084, pp. 132–146, Springer-Verlag, Berlin, 1996.

[10] D. R. Karger and C. Stein: An $O(V^2)$ algorithm for minimum cuts, *Proc.* 25*th Annual ACM Symp. on Theory of Computing*, pp. 757–765, 1993.

[11] H. Nagamochi and T. Ibaraki: Computing the edge-connectivity of multigraphs and capacitated graphs, *SIAM J. Discrete Math.*, 5: 54–66, 1992.

[12] H. Nagamochi and T. Ibaraki: A fast algorithm for computing minimum 3-way and 4-way cuts, *Proc.* 7*th Conf. on Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science 1610, pp. 377–390, Springer-Verlag, Berlin, 1999.

[13] H. Nagamochi, K. Nishimura, and T. Ibaraki: Computing all small cuts in an undirected network, *SIAM J. Discrete Math.*, 10(3): 469–481, 1997.

[14] H. Saran and V. V. Vazirani: Finding k cuts within twice the optimal, *SIAM J. Comput.*, 24(1): 101–108, 1995.

[15] V. Vazirani and M. Yannakakis: *Suboptimal Cuts*: *Their Enumeration, Weight, and Number*, Lecture Notes in Computer Science 623, pp. 366–377, Springer-Verlag, Berlin, 1992.