Journal of Algorithms **39**, 68–77 (2001)
doi:10.1006/jagm.2000.1148, available online at http://www.idealibrary.com on IDEAL®

A Simple Algorithm for the Planar Multiway Cut Problem

Wei-Chang Yeh

Department of Industrial Engineering, Feng Chia University, P.O. Box 67-100, Taichung, Taiwan 408, Republic of China
E-mail: wcyeh@fcu.edu.tw

Received August 18, 1999

The traditional min-cut problem involves finding a cut with minimum weight between two specified vertices. The planar multiway cut problem is a NP-hard generalization of the min-cut problem. It involves separating a weighted planar graph with k specified vertices into k components such that the total weight between the components is minimized. This problem has important applications in computer science, engineering, and management science. In this study, we developed a very simple algorithm with time complexity $O((k-\frac{3}{2})^{k-1}\cdot(n-k)^{2k-4}\cdot[nk-\frac{3}{2}k^2+\frac{1}{2}k]\cdot\log(n-k))$. Our algorithm is based on some simple theorems that characterize the structure of the k-way cut. It is also better than the best known algorithm with time complexity $O(4k^k\cdot n^{2k-1}\cdot\log n)$ for the planar multiway cut problem. © 2001 Academic Press

Key Words: planar graph; algorithm; cut; multiway cut.

1. INTRODUCTION

Consider a graph with weights assigned to its edges and a subset of the vertices called terminals. The multiway cut problem involves minimizing the sum of the weight set of edges that separates each terminal from all of the others. A variety of practical applications or potential relevance has been suggested [1], such as the minimization of communication costs in parallel computing systems, the clustering problem, VLSI design [12], and network reliability. Most of the literature associated with the multiway cut problem deals with the case where terminals are not specified [2–8]. In this paper, we focus on the multiway cut problem in a planar graph with k terminals and provide a new algorithm to find a minimal multiway cut in a planar graph.

If k is not fixed, the multiway cut problem for planar graphs is NP-hard even for edge weights equal to 1 [1]. Based on some of the special

0196-6774/01 $35.00

structures of the k-way cut, the proposed algorithm can be run in $O((k-\frac{3}{2})^{k-1}\cdot(n-k)^{2k-4}\cdot[nk-\frac{3}{2}k^2+\frac{1}{2}k]\cdot\log(n-k))$ time, which is simple and easy to implement and better than the best known algorithm [1, 9, 11].

The remainder of this paper is organized as follows. In Section 2, terminology and notation are defined. We present some necessary theorems and proofs in Section 3. Then in Section 4, we illustrate the details of our method and discuss its complexity. In Section 5, we present an example to demonstrate our algorithm. Concluding remarks are given in Section 6.

2. DEFINITIONS

The weighted graphs considered here are finite, undirected, simple, and planar. Standard terminology is used. All graphs are connected. Otherwise the multiway cut problem trivially reduces to smaller problems. The notation $G(V,E,W)$ denotes a planar graph with a set of vertices V, a set of edges E, and the weight function W. T denotes the set of terminals, $n=|V|$ the number of vertices, $m=|E|$ the number of edges, and $k=|T|$ the number of terminals. A *cut* is a partition of V into two parts S and $S^*=V-S$. Each cut defines a set of edges consisting of those edges that have one endpoint in S and another endpoint in S^*. A *min-cut* that separates a vertex s and a vertex t is a cut with minimal total weights of edges. The following notations are used: G^{D} is the dual graph of G, C is a multiway cut separating the graph into k components, $C^{\mathrm{D}}\subseteq G^{\mathrm{D}}$ is the dual multiway cut of C, S_i^{C} is the component containing terminal v_i after removing C, sub-cut $C_i^{\mathrm{C}}\subseteq C$ is the min-cut (the cut with minimal total weight) that separates terminal v_i from $T-\{v_i\}$, and $C_i^{\mathrm{CD}}\subseteq C^{\mathrm{D}}$ is the dual of C_i^{C}. If $C_i^{\mathrm{C}}\cap C_j^{\mathrm{C}}\neq\emptyset$, then S_i^{C} and S_j^{C} are adjacent to each other. It is trivially true that the dual of a cut in a planar graph is a cycle of the dual of this planar graph. For example, in the planar graph G of Fig. 1, $T=\{1,4,8\}$ is the terminal set. $C=\{e_{47},e_{54},e_{24},e_{35},e_{25}\}$ is a 3-way cut (see the dashed line in Fig. 1b), $C_1^{\mathrm{C}}=\{e_{24},e_{35},e_{25}\}$ is the subcut corresponding to C, and C_1^{CD} is the dual of C_1^{C}, i.e., a circle connecting nodes a, b, and c in Fig. 1d. S_1^{C} is the component containing terminal 1 after removing C (see Fig. 1c). G^{D} is the dual graph of G (see the dotted line in Fig. 1d).

A *face* of G is a region of the plane bounded by edges that satisfies the condition that any two points in the region can be connected by a continuous curve that meets no nodes and arcs. The *boundary* of a face, f, is the set of edges that enclose it. If the degree of a vertex of C^{D} in G^{D} is greater than or equal to 3, then such a vertex is called a *dual-joint* in C^{D}. A dual-joint in G^{D} is a face in G, and the boundary of such a face is called

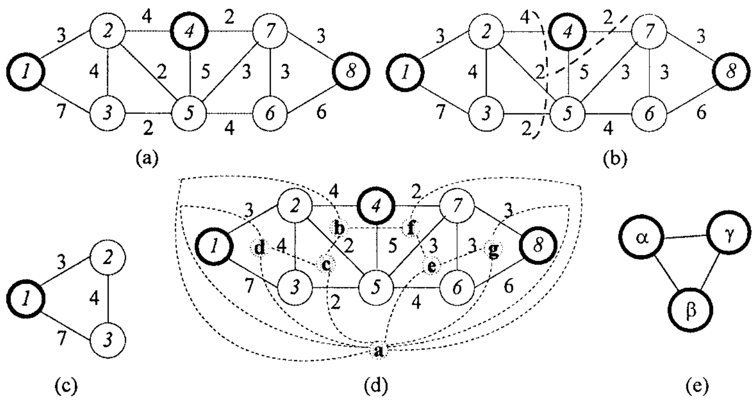

FIG. 1. An example graph.

a *joint-cycle*. For example, in Fig. 1d, the area inside $\{e_{24}, e_{45}, e_{52}\}$ is a face, node b is a dual-joint, and $\{e_{24}, e_{45}, e_{52}\}$ is a joint-cycle of node b.

The *component-graph* is a special graph obtained from the original graph G. Each component S_i in G is denoted by a vertex in the corresponding component-graph. If two components are adjacent, then the two corresponding vertices are connected by an edge in the component-graph. Furthermore, there is a corresponding face in the graph to each face in a component-graph. For example, the graph in Fig. 1e is a component-graph corresponding to the graph in Fig. 1b. If vertices α and β in Fig. 1e are corresponding to S_1^{C} (the component with respect to terminal 1) and S_2^{C} (the component with respect to terminal 4) in Fig. 1b, then the edge between vertices α and β is present and S_1^{C} and S_2^{C} are *adjacent*. The face around $\{e_{24}, e_{45}, e_{52}\}$ in Fig. 1b corresponds to the face inside Fig. 1e.

3. PRELIMINARIES AND PROPERTIES

We will first present some useful preliminaries and properties relating to the relationship among components, subcuts, and dual-joints. The following theorem relates to subcuts and components. It not only holds for any k-way cut (not necessarily a min k-way cut), but is also true for any nonplanar graphs.

LEMMA 1. *Let S_i^{C} and C_i^{C} be a component and a subcut of a min k-way cut C in a planar graph G, respectively, where $i = 1, 2, \ldots, k$. If S_i^{C} and C_i^{C} are removed from the graph, then $C' = C - C_i^{\mathrm{C}}$ is a min $(k$-1$)$-way cut of the updated graph.*

Proof. Let $k' = k - 1$, $T' = T - \{v_1\}$, and G^* be the updated graph after removing S_i^{C} and C_i^{C} from G. If C' is not a min k'-way cut separating all of the terminals in T' of G^*, then there exists a min k'-way cut C^* such that $W(C^*) < W(C')$. Let $C^{\#} = (C - C') \cup C^*$. Since C^* separates all of the terminals in T' and $C - C^*$ separates T' from v_1, then $C^{\#}$ is also a k-way cut. Since $(C - C') \cap C^* = \emptyset$, $C' \subset C$, and $W(C^*) < W(C')$, we have $W(C^{\#}) = W(C) - W(C') + W(C^*) < W(C)$; i.e., $W(C^{\#}) < W(C)$. This contradicts the fact that C is a min k-way cut. It now follows that C' is a min k'-way cut in G^*. ▮

From Lemma 1, if one component can be identified from the graph, then the remaining min k-way cut is a min (k-1)-way cut in the remaining graph after removing this component and its relative cut. If we repeatedly perform Lemma 1, we will have the following theorem.

THEOREM 1. *If we create one graph, G^{C}, by removing $\cup S_i^{\mathrm{C}}$ and $\cup C_i^{\mathrm{C}}$ from $G(V, E, W)$, then $C^* = C - (\cup C_i^{\mathrm{C}})$ is a min k'-way cut for G^*, where $i = 1, 2, \ldots, p$, $1 \le p < k$ and $k' = k - p$.*

The following corollary is immediate from Theorem 1.

COROLLARY 1. *After removing k-2 components and their subcuts of a min k-way cut from the graph, the updated min k-way cut is just a min-cut for the updated graph if the updated graph is still a connected graph.*

From Theorem 1, if we can repeatedly identify a component and remove it with its relative subcut from the remaining graph, then all of the k components can be found; i.e., a min k-way cut is found. However, it is very difficult to identify a component. But if a subcut is found, then its relative component is formed. Therefore, identifying a subcut is instead the focus of this problem. The following theorem discusses this point.

THEOREM 2. *Let C be a k-way cut and C_i^{CD} contain p dual-joints $\varpi_1, \varpi_2, \ldots, \varpi_p$. If $C_i^* \neq C_i^{\mathrm{C}}$ is a cut separating $\{v_i, v_{i1}^*, v_{i2}^*, \ldots, v_{ip}^*\}$ from $T - \{v_i\}$, then $C' = (C - C_i^{\mathrm{C}}) \cup C_i^*$ is also a k-way cut, where $v_{ij}^* \in V$ is a vertex in both S_i^{C} and the joint-cycle of ϖ_j, $i = 1, 2, \ldots, k$, and $j = 1, 2, \ldots, p$.*

Proof. Since C_{i}^* is a min-cut separating $\{v_i, v_{i1}^*, v_{i2}^*, \ldots, v_{ip}^*\}$ from $T - \{v_i\}$, there is no terminals connecting to v_i by a path after removing C'. For any subcut $C_j^{\mathrm{C}} \neq C_i^{\mathrm{C}}$ in C, C_j^{C} is also a subcut in C' that separates $\{v_j\}$ from $T - \{v_i\}$ while $C_j^{\mathrm{C}} \in C \cap C'$. Assume $C_j^{\mathrm{C}} \notin C \cap C'$; i.e., $C_i^{\mathrm{C}} \cap C_j^{\mathrm{C}} \neq \emptyset$. Thus there is at least a dual-joint, say ϖ_1, in C_j^{CD} and a cut $C_l^{\mathrm{C}} \in C$ such that $C_j^{\mathrm{C}} \cap C_l^{\mathrm{C}} \neq \emptyset$, $C_l^{\mathrm{C}} \cap C_l^{\mathrm{C}} \neq \emptyset$, and $\varpi_1 \in C_l^{\mathrm{CD}}$ according to the fact that the degree of a dual-joint is greater than or equal to 3. Consider the following two cases:

Case 1. If $\varpi_1 \in C_i^{*\mathrm{D}}$ (see Fig. 2a) and/or $C_i^* \cap C_j^{\mathrm{C}} \neq \emptyset$ (see Fig. 2b), then it is trivially true that there are no paths between v_i and v_l.

Case 2. If $\varpi_1 \notin C_i^{*\mathrm{D}}$ and $C_i^* \cap C_j^{\mathrm{C}} = \varnothing$ (see Fig. 2c), then it contradicts the fact that C_i^* is a cut between $\{v_i, v_{i1}^*, v_{i2}^*, \ldots, v_{ip}^*\}$ and $T - \{v_i\}$.

From the above, there is no path between any two terminals after removing C'. Therefore, C' is also a k-way cut. ∎

Following directly from Theorem 2, the following theorem summarizes the implication of the related result. It forms the basis for our algorithm.

THEOREM 3. *Let C be a min k-way cut. If C_i^{CD} contains p dual-joints $\varpi_1, \varpi_2, \ldots, \varpi_p$, then C_i^{C} is a min-cut separating $\{v_i, v_{i1}^*, v_{i2}^*, \ldots, v_{ip}^*\}$ from $T - \{v_i\}$ and $W((C - C_i^{\mathrm{C}}) \cap C_i^*) = \varnothing$, where $v_{ij}^* \in V$ is a vertex in both S_i^{C} and the joint-cycle of ϖ_j for any $C_i^* \neq C_i$ is a min-cut separating $\{v_i, v_{i1}^*, v_{i2}^*, \ldots, v_{ip}^*\}$ from $T - \{v_i\}$, $i = 1, 2, \ldots, k$, and $j = 1, 2, \ldots, p$.*

Proof. From Theorem 2, $C' = (C - C_i^{\mathrm{C}}) \cup C_i^*$ is also a k-way cut, and

$$\begin{aligned} W(C') &= W\big((C - C_i^{\mathrm{C}}) \cup C_i^*\big) \\ &= W\big(C - C_i^{\mathrm{C}}\big) + W(C_i^*) - W\big((C - C_i^{\mathrm{C}}) \cap C_i^*\big) \\ &< W\big(C - C_i^{\mathrm{C}}\big) + W(C_i^*) \\ &< W\big(C - C_i^{\mathrm{C}}\big) + W(C_i) \\ &= W(C). \end{aligned}$$

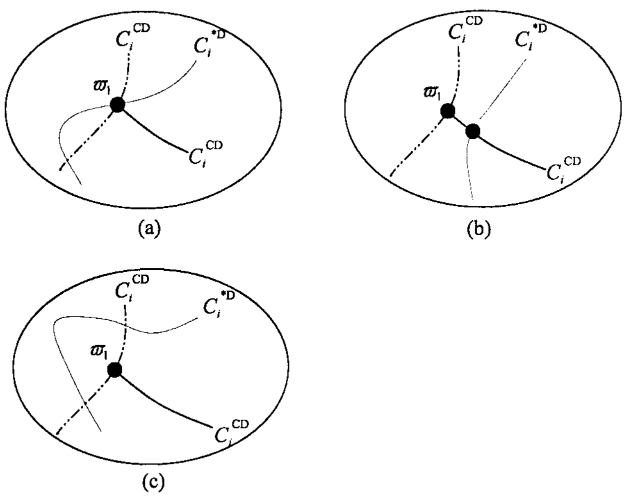

FIG. 2. Illustrating the possibilities of the relationship between $C_i^{*\mathrm{D}}$ and C_i^{CD}.

Thus, $W(C') < W(C)$. This contradicts the fact that C is a min k-way cut. The proof follows from above. ∎

Theorems 2 and 3 show that dual-joints determine subcuts. However, which nodes need to be chosen in Theorems 2 and 3 to find a subcut is not quite obvious. Otherwise, the min k-way cut problem can be solved in polynomial time. Fortunately, the maximum number of nodes required to be chosen in Theorem 3 can be predicted. First, a special structure related to the component-graph is discussed in Theorem 4. Through Theorem 4, the maximum number of nodes required to be chose is proposed in Theorem 5.

THEOREM 4. *A component-graph is a planar graph.*

Proof. The component-graph is obtained from the original graph G by shrinking each connected component S_i into a single vertex and then deleting self-loops and multiple edges, where $i = 1, 2, \dots, k$. These operations do not destory planarity. ∎

The following lemma states some fundamental properties about the planar graph. Its proof can be found in any book about graph theory.

LEMMA 2. *Let n be the number of vertices, f be the number of faces, and m be the number of edges in a planar graph. Then $n + f - 2 = m \leq 3n - 6$ and $f \leq 2n - 4$.*

The following theorem is a very important theorem in our paper which describes the fact that the total number of dual-joints of any planar k-way cut is equal to or less than the total number of the faces in the component-graph, i.e., $2k - 4$.

THEOREM 5. *There are at most $2k - 4$ dual-joints for every planar k-way cut.*

Proof. There are k vertices in the component-graph. Hence, if f is the number of faces in the component-graph, then $f \leq 2k - 4$ by Lemma 2. Besides, from the property of dual-joints, the dual-joint of a dual graph is in the face of the graph. Thus, there are at most $2k - 4$ dual-joints for every planar k-way cut. ∎

The following corollary follows directly from Theorem 5.

COROLLARY 2. *There are at most $2k - 4$ dual-joints in C_i^{CD}.*

4. ALGORITHM AND COMPLEXITY

We now state our algorithm and establish its complexity. From Theorem 3, choose a terminal in T, say v_1, and m_1 nonterminals. Then find a min-cut separating v_1 and these m_1 nonterminals from $T - \{v_1\}$, where $0 \le m_1 \le 2k - 4$. Then replace G by removing this min-cut and the component w.r.t. v_1 from G, and replace T by $T - \{v_1\}$. In the same way, we choose another terminal, say v_2, and m_2 nonterminals to find a min-cut separating v_2 and these m_2 non-terminals from $T - \{v_2\}$, where $0 \le m_2 \le \min\{2[(k-2)-1], 2k-4-m_1\}$. Then replace G by removing this min-cut and the component w.r.t. v_2 from G, and replace T by $T - \{v_2\}$. By repeatedly using the same procedure to find the min $|T|$-way cut in G and trying all of the above combinations, we will have a min k-way cut. From our procedure, the limited enumeration of all possible k-way cuts and the evaluation of the weight of the cut for each one will certainly yield the optimal solution.

To simplify the computation of the complexity and notation, let $n_i = n - k - \sum_{j=1}^{i} m_j = n_{i-1} - m_i$ and $n_0 = n - k$, and m_i is the possible number of nonterminals that need to be chosen to remove the ith component, where $i = 1, \dots, k-1$. From Theorem 5 and Corollary 1, $\sum_{j=1}^{k-1} m_j \le 2k - 4$ and $0 \le m_i \le 2k - 4$, where $i = 1, \dots, k-1$. Thus, the total number of combinations of the values of $m_1, m_2, \dots, m_{k-1}$ is $(2k - 4 + 1)^{k-1} = (2k-3)^{k-1}$.

It takes $O(n \cdot \log n)$ time to find a min-cut in the planar graph with n vertices [10]. Therefore, after assigning the nonterminals to each terminal, it takes $[n_0 \log n_0 + (n_1 - 1)\cdot\log(n_1 - 1) + \cdots + (n_{k-1} - 1)\cdot\log(n_{k-1} - 1)]$ time to find such k-way cut. Furthermore, the total complexity is a product of the complexity in each procedure to find a k-way cut:

$$
\begin{aligned}
&\frac{n_0!}{m_1! m_2! \cdots m_{k-1}! n_{k-1}!} \\
&\qquad \cdot\left[n_0 \log n_0 + (n_1 - 1)\cdot\log(n_1 - 1) + \cdots \right. \\
&\qquad\qquad \left. + (n_{k-1} - 1)\cdot\log(n_{k-1} - 1)\right] \\
&\quad \le \frac{(n-k)!}{2!2!\cdots 2![(n-k) - (2k-4)]!} \\
&\qquad \cdot\left[n_0 + (n_1 - 1) + \cdots + (n_{k-1} - 1)\right]\cdot\log n_0 \\
&\quad \le \frac{(n-k)^{2k-4}}{2^{k-1}}\cdot\left[nk - \frac{3}{2}k^2 + \frac{1}{2}k\right]\cdot\log(n-k).
\end{aligned}
$$

From above, the total time complexity of all possible enumerations for different choices of $m_1, m_2, \ldots, m_{k-2}$ is

$$\sum_{m_1, m_2, \ldots, m_{k-1}} \frac{n_0!}{m_1! m_2! \cdots m_{k-1}! n_{k-1}!}$$

$$\cdot [n_0 \log n_0 + \cdots + (n_{k-1} - 1) \cdot \log(n_{k-1} - 1)] \cdot \log(n-k)$$

$$\le (2k-3)^{k-1} \cdot \frac{(n-k)^{2k-4}}{2^{k-1}} \cdot \left[nk - \frac{3}{2}k^2 + \frac{1}{2}k\right] \cdot \log(n-k)$$

$$\le \left(k - \frac{3}{2}\right)^{k-1} \cdot (n-k)^{2k-4} \cdot \left[nk - \frac{3}{2}k^2 + \frac{1}{2}k\right] \cdot \log(n-k),$$

where $n_i = n - k - \sum_{j=1}^{i} m_j = n_{i-1} - m_i$, $n_0 = n - k$, and $0 \le m_i \le 2k - 4$, $\forall\ i = 1, 2, \ldots, k-1$.

In the above analysis, to estimate the complexity, we assume that every component we remove has only one terminal and related dual-joints. Of course, this assumption is not correct because the vertices in each component contain more than one terminal and related dual-joints. The real running time is smaller than the overestimated time complexity.

Since $(k - \frac{3}{2})^{k-1} \cdot (n-k)^{2k-4} \cdot [nk - \frac{3}{2}k^2 + \frac{1}{2}k] \cdot \log(n-k) \ll 4k^4 \cdot n^{2k-1} \cdot \log n$, our algorithm is better than the best known algorithm proposed by Dahlhaus *et al.* [1] with a time complexity of $O(4k^4 \cdot n^{2k-1} \cdot \log n)$. Although this algorithm is simpler than the existing algorithms [1, 9], there is no dual graph for a nonplanar graph. Therefore, it also works only on a planar graph as does the best-known algorithm proposed in [1].

5. AN EXAMPLE

We present an example to illustrate the proposed algorithm.

EXAMPLE 1. Find the min 3-way cut in the planar graph in Fig. 1a, where vertices 1, 4, and 8 are terminals.

Solution

Part 1 (S_1 without dual-joint): $\{e_{24}, e_{25}, e_{35}\}$ is a min-cut between 1 and $\{4, 8\}$. After removing this min-cut, one of the min-cuts between 4 and 8 is $\{e_{45}, e_{47}\}$. Thus the 3-way cut is $\{e_{24}, e_{25}, e_{35}, e_{45}, e_{47}\}$, and the weight is $W(e_{24}) + W(e_{25}) + W(e_{35}) + W(e_{45}) + W(e_{47}) = 15$.

Part 2 (S_1 with 1 dual-joint): $\{e_{24}, e_{25}, e_{35}\}$ is a min-cut between $\{1, 2\}$ and $\{4, 8\}$. After removing such a min-cut, $\{e_{78}, e_{68}\}$ is a min-cut between terminals 4 and 8. So, in this case, the total weight of this 3-way cut is $W(\{e_{24}, e_{25}, e_{35}\} \cup \{e_{47}, e_{45}\}) = 15$. If we try $\{1, 3\}$, we will produce the same answer as above, 15. If we try $\{1, 5\}$, $\{1, 6\}$, and $\{1, 7\}$, we still cannot find a 3-way cut with a weight less than 15 (up to now, 15 is the lower bound of any min 3-way cut).

Part 3 (S_1 with 2 dual-joints): If we try $\{1, 2, 3\}$, $\{1, 2, 5\}$, $\{1, 2, 6\}$, $\{1, 2, 7\}$, $\{1, 3, 5\}$, $\{1, 3, 6\}$, $\{1, 3, 7\}$, $\{1, 5, 6\}$, $\{1, 5, 7\}$, and $\{1, 6, 7\}$, we still cannot find a 3-way cut with a weight less than 15. For example, $\{e_{24}, e_{25}, e_{35}\}$ is a min-cut between $\{1, 2, 3\}$ and $\{1, 4\}$. After removing this min-cut, $\{e_{47}, e_{45}\}$ is a min-cut between terminals 4 and 8. The total weight of this 3-way cut is $W(\{e_{24}, e_{25}, e_{35}\} \cup \{e_{47}, e_{45}\}) = 15$.

From the above, we know that the min 3-way cut is $\{e_{47}, e_{54}, e_{24}, e_{35}, e_{25}\}$ and that the weight of the min 3-way cut is 15.

6. CONCLUSIONS

The problem of partitioning or clustering some of the specified vertices in a weight planar graph, with unspecified size, has been of interest to OR researchers for many years. This problem arises in various physical situations. Unfortunately, the problem is a NP-hard generalization of the min-cut problem.

In this study, we developed an efficient and simple algorithm for the multi-way cut problem. The best known algorithm for this problem was proposed by Dahlhaus *et al.* [1] with a complexity of $O(4k^4 \cdot n^{2k-1} \cdot \log n)$. Our method, with a time complexity of $O((k - \frac{3}{2})^{k-1} \cdot (n - k)^{2k-4} \cdot [nk - \frac{3}{2}k^2 + \frac{1}{2}k] \cdot \log(n - k))$, is not only more efficient than the best known algorithm, but also simpler, because it uses only fundamental concepts from graph theory to characterize the structure of the k-way cut to limit the size of the subsets enumerated instead of a straightforward enumeration procedure.

From our procedure, the limited enumeration of all possible k-way cuts and evaluation of the weight of each cut will certainly yield the optimal solution. However, the number of possible k-way cuts is exponential in n. Therefore, there is still ample improvement possible in the run time for an algorithm solving the k-way cut problem.

ACKNOWLEDGMENTS

I thank both the associate editor and the referees for their constructive comments and recommendations, which have significantly improved the presentation of this paper. This research was supported in part by the National Science Council of Taiwan, R.O.C. under Grant NSC 82-0113-E-214-027-T.

REFERENCES

1. E. Dalhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, The complexity of the multiway cuts, *SIAM J. Comput.* **23** (1994), 864–894.
2. O. Goldschmidt, "Deterministic and Probabilistic Aspects of the k-Way Cut Problem," Ph.D. Dissertation, U.C. Berkeley, 1988.
3. O. Goldschmidt and D. S. Hochbaum, Polynomial algorithm for the k-way cut problem, *in* Proceedings, 29th IEEE FOCS, 1988.
4. O. Goldschmidt and D. S. Hochbaum, "A Polynomial Algorithm for the k-Way Cut Problem for Fixed k," Unpublished paper, 1991.
5. O. Goldschmidt and D. S. Hochbaum, Asymptotically optimal linear algorithm for the minimum k-way cut in a random graph, *SIAM J. Discrete Math.* **3** (1990), 58–73.
6. X. He, An improved algorithm for the planar 3-way cut problem, *J. Algorithms* **12** (1991), 23–37.
7. D. S. Hochbaum and D. B. Shmoys, An $O(|V|^2)$ algorithm for the planar 3-way cut problem, *SIAM J. Algebraic Discrete Math.* **6** (1985), 707–712.
8. D. S. Hochbaum and L. Tsai, "A Greedy Algorithm for the 3-Way Cut Problem and Its Worse Case Bound," Technical report IP-318, Economic Theory and Econometrics, U. C. Berkeley, 1983.
9. H. S. Wang, "On Minimum-Fold Cuts in Networks," Ph.D. Dissertation, University of Texas at Arlington, 1990.
10. J. Hopcroft and R. Tarjan, Algorithm 447: Efficient algorithms for graph manipulation, *Comm. Assoc. Comput. Mach.* **16** (1973), 327–378.
11. W. C. Yeh, M. S. Yeh, and H. W. Corley, A polynomial algorithm for the planar k-way cut problem, *in* "CIIE National Conference," pp. 58–63, 1992.
12. H. S. Stone, Multiprocessor scheduling with the aid of network flow algorithms, *IEEE Trans. Software Engrg.* **3** (1997), 85–93.