Journal of the Korean Institute
of Industrial Engineers
Vol. 19, No.3, September., 1993.

CUT TREE의 재구축

RECONSTRUCTION OF THE CUT TREE

김 채 복*

Chae-Bogk Kim*

Abstract

This paper develops $0(n^3)$ algorithm to construct a cut-tree generated by Gomory-Hu algorithm. The algorithm only requires node sets defined by the minimal cut in each of the (n-1) maximal flow determinations. Merging computerized facility layout procedure that uses cut-tree concept to generate design skeletons with our algorithm requires less storage space than merging it with Gomory-Hu algorithm. Also, the cut-tree can easily be modified when the (n-1) minimal cut-sets are updated due to changes on arc capacities.

1. INTRODUCTION

Consider a capacitated undirected network where a commodity is transported from any node to any other node in the network. A classical example is a transportation network where each node generates flow and receives flow at different points in time. The amount to be transported is only limited by the capacities of the arcs (distribution links) in the network. For such networks, analysts are often interested in finding maximal flow between all pairs of nodes in the network. Gomory-Hu algorithm [2] solves this problem in (n-1) maximum flow determinations, where n is the number of nodes in the network. The basic idea of the method is to successively construct a maximal spanning tree with arcs representing cuts and the arcs paramters representing the value of the cut separating the nodes in the subsets formed by deleting the arc from the tree. The tree generated is referrned to as cut-tree since the arcs of the tree refers to cuts in the original network.

Although one can use the Gomory-Hu algorithm to find the cut tree, as Gusfield [3] states, it is not a trivial algorithm to program. Gusfield [3] provides a simple and efficient technique to generate the cut tree, not restricting his method to noncrossing cuts of the network.

Adolphson and Hu [1] discuss a backboard

* 고려대학교 공과대학 산업공학과

wiring problem where n pairs connected by wires should be placed into n holes (all in a line) such that the total wire length is minimum. They have shown that if the Gomory-Hu cut-tree is a chain, then it is the optimal solution to the problem. Otherwise the two most extreme nodes of the cut-tree are in the optimal linear ordering.

Although graph theoric approaches have been applied effectively for layout problems, the concentration has mainly been on adjacency matrix information rather than the traffic between two nodes. Recently, cut-trees have been using as design skeletons for facility layout problems [4]. Given the material flow matrix, a cut-tree is generated using Gomory-Hu algorithm to determine the maximal flow between every node pair. The cut-tree provides information such as which departments should be separated in the layout.

One actually does not need to generate the cut-tree explicitly while using Gomory-Hu algorithm. The following procedure generates maximum flow between all node pairs without explicitly generating the cut-tree. A cut-set C(X, X') separates the nodes into two disjoint node sets, X and X'. At each step two nodes from the some node set are used as the source and sink node for the maximum flow calculations. The newly generated cut-set further breaks down the node set into two smaller node sets. Eventually, every node set contains only one node signaling the termination of the process. At this point, maximum flow values between (n-1) node pairs and the respective cut-sets are on hand. There remains to determine maximum flow between the remaining node pairs. The maximum flow between node i and node j not included in (n-1) maximum flow determinations cna be found as follows. Let X_{kl} *and* X'_{kl} and denote the two node sets generated by the minimal cut separating node k (source node) and node l (sink node) generated after maximum flow determination. Let V_{kl} be the corresponding minimal cut, hence, maximal flow value. Define,

$S=\{[k,l]\} : i \in X_{kl},\ j \in X'_{kl},\ or\ i \in X_{kl}, j \in X_{kl}\}$

Thus, $v_{ij}=\min\{[k,l]\in s\}\ (v_{kl})$. Hence, the above procedure generates the maximum flow matrix without explicitly making use of the cut-tree. Certain class of problems such as layout problems require construction of a cut-tree from maximal flow matrix. Although, it is somewhat easier for a human to generate the cut-tree, one needs a step-by-step procedure which can be easily computerized.

In this paper, we present an efficient algorithm to reconstruct the cut-tree from the (n-1) cut-sets generated after (n-1) maximum flow determinations. Section 2 develops the algorithm for reconstructing the cut-tree. Section 3 analyzes the performance of the algorithm. Conclusions are given in section 4.

2. RECONSTRUCTING A CUT-TREE

Let G=(N, A) be an undirected network where N and A are the node and arc sets, respectively. Let c_{ij} be the capacity of the arc (i,j), by symmetry $c_{ij}=c_{ji}$. The maximum flow from node i to node j is indicated by v_{ij}. The corresponding minimal cut-set and the two node set are denoted by $C(X, X')_{ij}$, X_{ij} and $\{X'\}_{ij}$, respectively. The Gomory-Hu algorithm generates (n-1) cut-sets. Define,

$t_k[i,j]=1$ if $k \in X_{ij}$
$=0$ otherwise

Define $F=\{ [i, j] \}$ as the set of source and sink pairs used for (n-1) maximum flow calculations. Brackets are used instead of parenthesis to separate source-sink pairs from arc representations. For any two nodes $u \in N$ and $v \in N$, define

$d_{uv}[i,j]=1$ if $t_u[i,j] \neq t_v[i,j]$
$=0$ otherwise

Lemma 1
The number of arcs between any two nodes u and v in the cut-tree is given by D_{uv} where

$D_{uv}=\sum_{[i,j]\in F} d_{uv}[i,j]$

(Proof) Suppose maximum flow from node i to node j is determined. For nodes u and v which are both either in X_{ij} or X'_{ij}, $t_u[i,j]=t_v[i,j]$ and hence $d_{uv}[i,j]=0$. For nodes $u \in X_{ij}$ and $w \in X'_{ij}$, $t_u[i, j] \neq t_w[i, j]$ and hence $d_{uw}[i, j]=1$. The arc added to the cut-tree at this step separates nodes in X_{ij} from nodes in X'_{ij} (Figure 1). Therefore, the added arc increases the number of arcs between any node in X_{ij} from nodes in X'_{ij} in the cut-tree by one. Next, maximum flow between node k and node l both in X_{ij} or both in X'_{ij} is determined. Without loss of generality, assume $k \in X_{ij}$ and $l \in X_{ij}$ are selected. Then, minimum cut $C(X,X')_{kl}$ further divides X_{ij} into two node sets with either $X'_{ij} \subset X'_{kl}$(Figure 2.a) or $X'_{ij} \subset X_{kl}$ (Figure 2.b). With a little effort, summation of $d_{uv}[i,j]$ over all maximum flow computations (that is, for all $[i,j] \in F$) establishes the desired result. ■

Lemma 2
The maximum flow between any two nodes u and v in the cut-tree is given by M_{uv} where

$M_{uv}= \min_{[i,j]\in F}\{v_{ij} : t_u[i,j] \neq t_v[i,j]\}$

(Proof) Maximum flow between any two departments is deqfined by finding a path (i.e., a path between any two nodes in a tree is unique) in the cut tree and selecting the minimum cut value, v_{ij}, corresponding to the path. Whenever $t_u[i,j] \neq t_v[i,j]$,

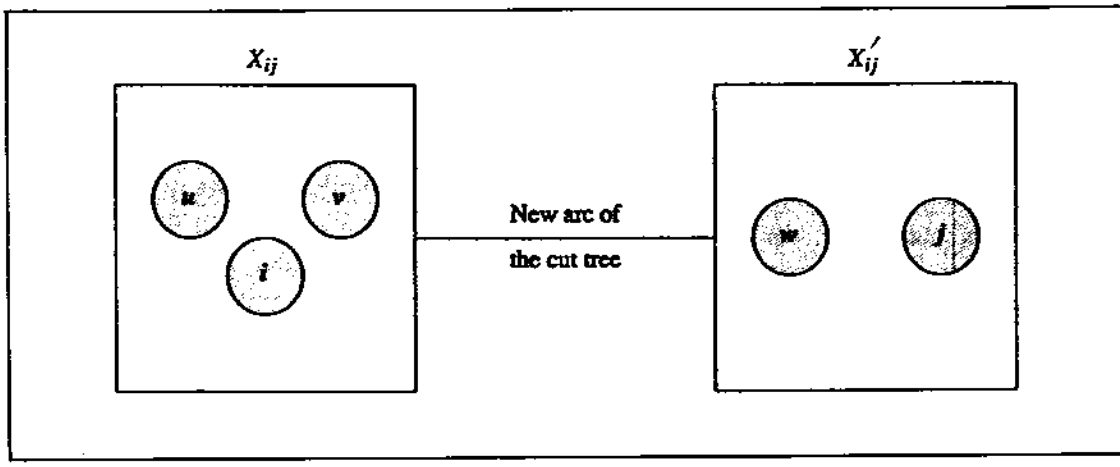

Figure 1. Expanding the cut tree by an arc

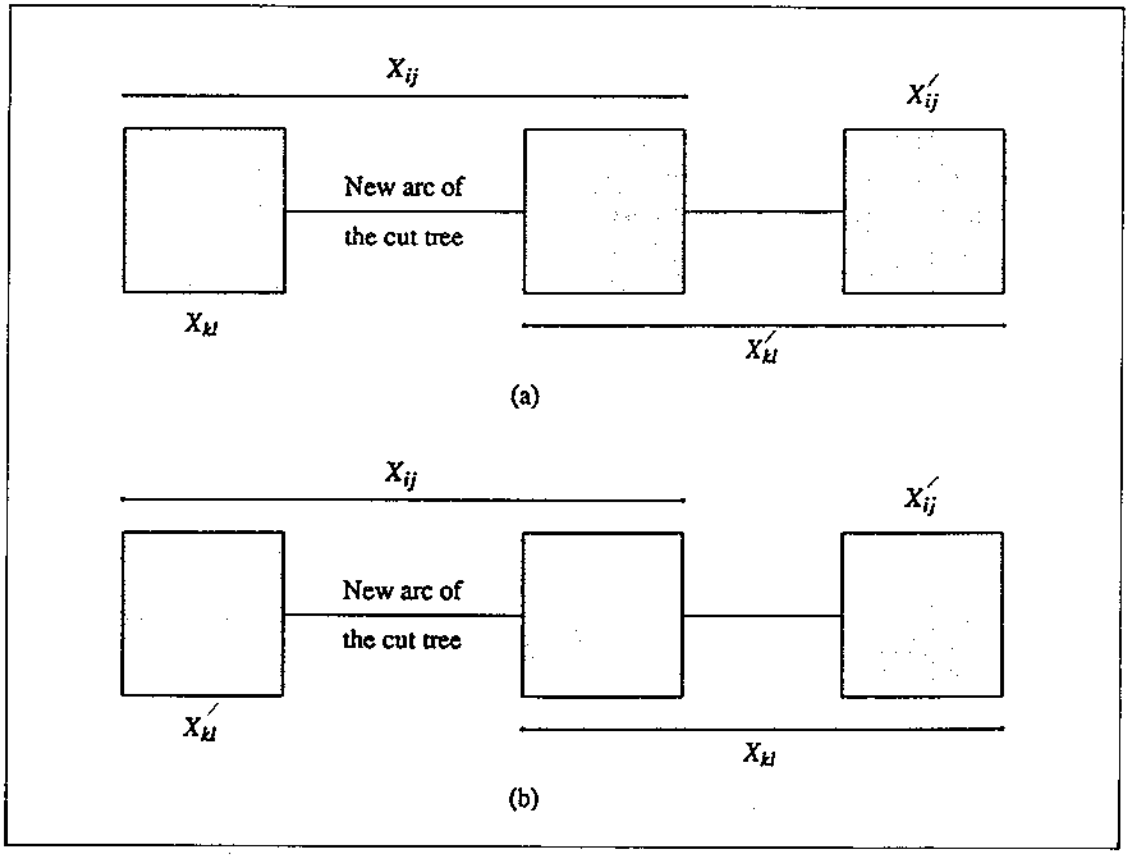

Figure 2. Expanding the cut tree by a new arc

the arc number of path between nodes u and v increases by one. Becaus of Lemma 1, finding the minimum cut value of the path between nodes u and v is equivalent to picking the minimum cut value, where $t_u[i,j] \neq t_v[i,j]$. ■

In the Gomory-Hu algorithm, some nodes are condensed, if necessary, in order to avoid crossing cuts. But, the crossing cut problem is not needed to consider in this paper due to Gusfield [3]. He presents a simple algorithm generating noncrossing cuts by adding just five or ten lines in the Gomory-Hu algorithm. The matrix $D = [D_{ij}]$ is a symmetric matrix with diagonal elements equal to zero. The following algorithm capitalized on the D matrix in generating the cut-tree for the problem. The inputs to the algorithm are X_{ij} and X'_{ij} sets for all $[i,j] \in F$.

ALGORITHM

Step1 : For each $[i,j] \in F$
If node $k \in X_{ij}$, set $t_{ij}(k) = 1$.
Otherwise, set $t_{ij}(k)$ 0.
Next[i,j]

Step2 : For each $(u, v) \in A$, $u < v$.
$D_{uv} = 0$.
For each $[i,j] \in F$
If $t_{ij}(v)$, go to Next [i,j].
Otherwise, set d_{uv} [i,j] =1
and $D_{uv} = D_{uv} + 1$.

Next [i,j]
$D_{uv} = D_{uv}$
Next (u,v)

Step 3 : Select any node u ∈ N as the root node of the cut-tree.
$N_0 = \{u\}$, $N' = N'-\{u\}$.
Draw node u.
w = 1.

Step 4 : B = ϕ.
For each i ∈ N_{w-1}
Determine $B_i = \{j : D_{ij} = 1$ and $D_{uj} = w\}$.
If $B_i = \phi$, go to Next i.
Draw nodes in B_i.
Draw arc (i,j) from node i to each node j ∈ B_i.
$B \leftarrow B + B_i$.
Next i

Step 5 : $N_w \leftarrow B$, $N' = N'-B$.
If $N' = \phi$, stop.
Otherwise, w=w+1 and return to Step 4.

The first two steps generate the distance matrix D, where D_{uv} gives the number of arcs of the path from node u and from node v in the cut-tree. Any node can be picked as the root node of the cut-tree. The chosen node is indicated by u. All those nodes which can be reached from node u using one arc define the first level of the cut-tree. In general, w refers to the level of the cut-tree currently under consideration. The set of nodes in level w which are incident to node i in level (w-1) is given by B_i. Note that, $B_i \in N_{w-1}$ are mutually exclusive sets, since otherwise the cut-tree will contain cycles. Step 4 is repeated until $N' = \phi$.

Example. Consider a flow matrix in Table 1 taken from [5]. With this flow matrix we can draw a network shown in Figure 3. The numbers assigned to the arcs on the network represent flow capacities. Table 2 shows source nodes sink nodes and corresponding cut values of the iterations when Gomory-Hu algorithm is employed. In order to obtain the cut tree grapph (Figure 4), three operations are needed : operation for node separation matrix (Table 3), operation for distance matrix (Table 4) and operation for maximum flow matrix (Table 5). We can obtain the node separation matrix at Step 2 in the proposed algorithm. From the node separation matrix, the distance matrix and the maximum flow matrix can be generated using Lemma 1 and Lemma 2, respectively.

Table 1. Flow matrix

D	1	2	3	4	5	6	7
1	0	8	9	7	0	0	0
2	8	0	0	5	7	0	0
3	9	0	0	4	0	9	0
4	7	5	4	0	4	6	8
5	0	7	0	4	0	0	2
6	0	0	9	6	0	0	11
7	0	0	0	8	2	11	0

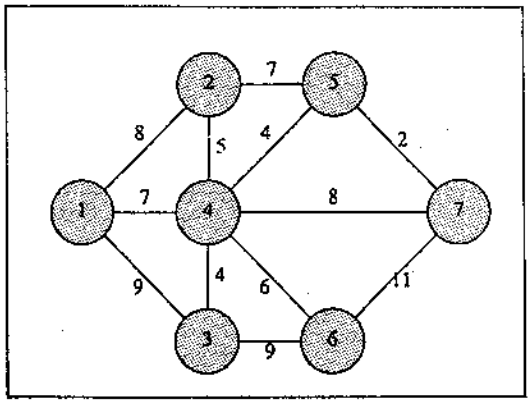

Figure 3. Multiterminal network

Table 2. Illustration of Gomory-Hu algorithm

iteration	source node	sink node	cut value
1	2	5	13
2	1	2	19
3	6	7	21
4	4	6	25
5	1	4	24
6	3	4	22

Table 4 Distance Matrix

D_{ij}	1	2	3	4	5	6	7
1	0	2	2	1	3	2	3
2	2	0	2	1	1	2	3
3	2	2	0	1	3	2	3
4	1	1	1	0	2	1	2
5	3	1	3	2	0	3	4
6	2	2	2	1	3	0	1
7	3	3	3	2	4	1	0

Table 3. Node separation matrix

source	sink	1	2	3	4	5	6	7
2	5	1	1	1	1	0	1	1
1	2	1	0	1	1	0	1	1
6	7	1	1	1	1	1	1	0
4	6	1	1	1	1	1	0	0
1	4	1	0	0	0	0	0	0
3	4	0	0	1	0	0	0	0

Table 5. Maximal flow matrix

M	1	2	3	4	5	6	7
1	0	19	22	24	13	24	21
2	19	0	19	19	13	19	19
3	22	19	0	22	13	22	21
4	24	19	22	0	13	25	21
5	13	13	13	13	0	13	13
6	24	19	22	25	13	0	21
7	21	19	21	21	13	21	0

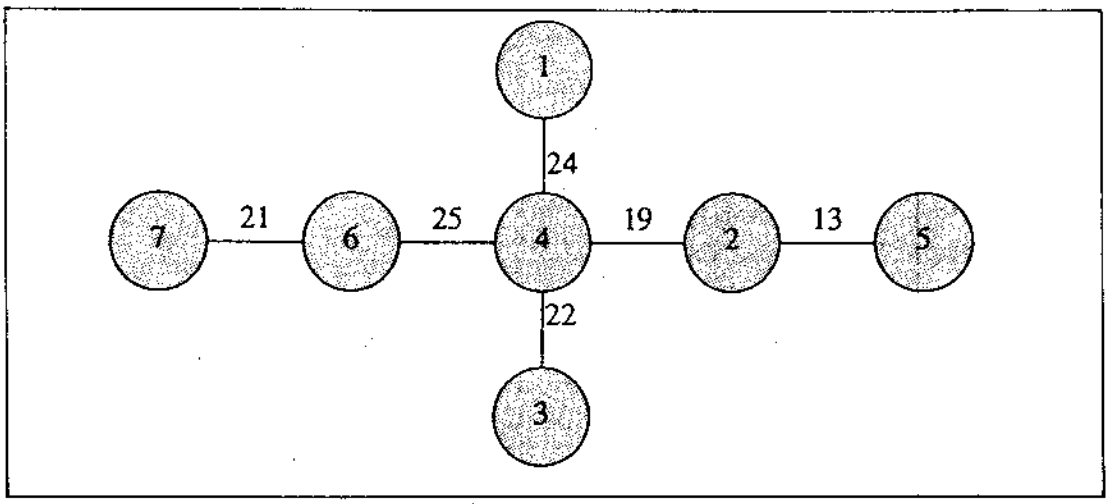

Figure 4. Cut tree

3. PERFORMANCE OF THE ALGORITHM

The worst case performance of the algorithm can be determined as follows. Step 1 is executed (n-1) times. In each repetition $t_{ij}(k)$ values are set for n nodes. Hence, Step 1 requires $0(n^2)$ time. Step2 requires (n-1) comparisons n(n-1)/2 times, hence 0 (n^3) time. Steps 4 and 5, together require 0 (n^2) time. Hence, theoretical complexity of

the procedure is $O(n^3)$. However, since at each step either simple comparisons or incrementations are done, the computational time of the algorithm is expected to be short.

For a problem such as a facility where the cut tree is required for an initial layout, a merge of proposed algorithm with a layout improvement technique will lead to consider able savings in computational time and required memory size. Also, we can use the distance matrix and maximum flow matrix for the facility layout problem when the cut tree approach [4] is employed. They provide a good criterion for the neighbor set in layout problem.

4. CONCLUSIONS

This paper presents an algorithm for reconstructing a cut-tree from (n-1) maximum flow determinations. The theoretical complexity is shown to be $O(n^3)$.

Given a change in X_{ij}, X'_{ij} for $[i,j] \in F$ due to change in arc capacities, one can update the cut-tree very easily using the concepts developed in this paper. The algorithm also is essential for researchers in the facility planning area if their designs are based on flow between any two departments in the facility.

Final remark concerning the problem is the adjacency information. The reconstruction of the cut-tree only requires X_{ij}, X'_{ij} sets for all $[i,j] \in F$ and not any specific information abcut the original network such as adjacency information on arcs. Hence, for facility layout problems, for example, where disk space is limited due to the size of the problem, one can run Gomory-Hu algorithm separately and store only X_{ij}, X'_{ij} for $[i,j] \in F$. This information is the only information required for a computerized facility layout procedure that uses cut-tree concepts.

References

1. Adolphson, D. and Hu, T.C., "Optimal Linear Ordering", SIAM J. APPL. Math., Vol.25-3, pp.403－423, 1973.
2. Gomory, R.E. and Hu, T.C., "Multi-Terminal Network Flows", J.SIAM, Vol.9, pp.551－1961.
3. Gusfield, D., "Very simple methods for all pairs network flow analysis", SIAM Journal on Computing, Vol.19, pp.143-155, 1990.
4. Montreuil, B. and Ratliff, H.D., "Utilizing Cut Trees as Design Skeletons for Facility Layout", IIE Transactions, Vol.21, pp. 136-143, 1989.
5. Phillips, Don T. and Garcia-Diaz Alberto, Fundamentals of Network Analysis, Prentice-Hall, INC., N.J. 1981.