1

ORC 67-22
JUNE 1967

SENSITIVITY ANALYSIS IN MULTITERMINAL FLOW NETWORKS

by

Francisco K. Rado

OPERATIONS RESEARCH CENTER

COLLEGE OF ENGINEERING

UNIVERSITY OF CALIFORNIA-BERKELEY

50

SENSITIVITY ANALYSIS IN MULTITERMINAL FLOW NETWORKS

by

Francisco K. Rado
Operations Research Center
University of California, Berkeley

June 1967 ORC 67-22

This research has been partially supported by the U.S. Army Research Office-Durham, Contract DA-31-124-ARO-D-331, and the National Science Foundation under Grant GK-1684 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

ABSTRACT

Algorithms are given for determining the influence of varying the capacity of an arc or of several arcs with linearly related capacities, on the maximal flows between every pair of nodes in an undirected network. This work builds upon the fundamental work of Gomory and Hu, and is closely related to the approach of Elmaghraby, who also gives algorithms for solving these problems.

SECTION I

Description of the Problem

Given a network G in which all arcs have finite capacities, a standard computational procedure due to Ford and Fulkerson [1] yields the maximal flow from one node called the source to another called the sink.

The problem of determining the maximal flows between every pair of nodes in an undirected network is called the maximal flow multi-terminal problem. Although there may be many pairs of nodes, there are at most n-1 different maximal multi-terminal flows, where n is the number of nodes in G .

The first problem solved in the present work is that of determining how varying of the capacity of an arc in a given range affects the n-1 values of the maximal flows. The second one generalizes the first by allowing the capacities of any subset of arcs of the network to depend linearly on a parameter constrained to a specified range, and then asks for the relation between the parameter values and the maximal flow values.

Briefly, the first problem is the sensitivity analysis of multi-terminal flow networks to changes in the capacity of one arc. The second extends this analysis to several arcs for a special kind of network; namely, one in which the capacities of the arcs are linearly related. Our chief result is the derivation of two algorithms that solve these problems.

Introductory Section

Because the fundamental results concerning maximal flows in networks are well known (see, e.g., [1]), we will review them here only to the extent necessary to establish the notation and definitions useful to this paper. Let $G = G(N,A)$ denote a network where N is the set of nodes and A the set of arcs of G. The arc a_{ij} joining node $\underline{i}$ to node $\underline{j}$ has capacity C_{ij} if the maximum amount of flow that this arc can carry from node $\underline{i}$ to node $\underline{j}$ is C_{ij}.

If the arc can carry a flow of C_{ij} units either from $\underline{i}$ to $\underline{j}$ or from $\underline{j}$ to $\underline{i}$ it is said that the arc is undirected. If every arc of A has that property, we speak of an undirected network.

Maximum flow - Minimum cut Theorem, Corollaries and algorithms related to it:

A cut C in $G(N,A)$ separating nodes $\underline{s}$ and $\underline{t}$ is a set of arcs $\{a_{ij}/i\varepsilon X, j\varepsilon\bar{X}\} = (X,\bar{X})$ where $\underline{s} \varepsilon X$ and $\underline{t} \varepsilon \bar{X}$. It also partitions the nodes in two subsets: X and $\bar{X}$. The capacity $C(X,\bar{X})$ of the cut $(X,\bar{X})$ is the sum of the capacities of the arcs of $(X,\bar{X})$.

The maximal flow - minimum cut theorem states:

For any network the maximal flow $\underline{f}$ from $\underline{s}$ to $\underline{t}$ is equal to the minimal cut capacity of all cuts separating $\underline{s}$ and $\underline{t}$.

A path from node X_1 to node X_n is a sequence of arcs $\{(X_k, X_{k+1})\}$, $k=1, \ldots, n-1$, where $X_k \neq X_p$ if $k \neq p$.

Some corollaries of the theorem, for undirected networks, are:

A cut $(X,\bar{X})$ is minimal if and only if every maximal flow $\underline{f}$ saturates all arcs of $(X,\bar{X})$ with flows going from X to $\bar{X}$. A flow $\underline{f}$ is maximal if and only if there is no flow augmenting path with respect to $\underline{f}$.

The algorithms for this problem are systematic searches of paths from source to sink, such that through each path we can carry a new flow given by the minimum of the differences of the capacities of the arcs of the path and the flows, in the direction of the path, already carried by them.

Multi-terminal network flows:

Sometimes the knowledge of the maximal flow between several or all pairs of nodes can be very useful. By naming any 2 nodes $\underline{i}$ and $\underline{j}$ source and sink, we can find the maximal flow f_{ij} between $\underline{i}$ and by the standard algorithms. Applying this procedure $\frac{n}{2}(n-1)$ times, (the number of combinations of two nodes in n) yields a maximal flow between every pair of nodes. (The maximal flow limited by capacity between $\underline{i}$ and $\underline{j}$ is the same as the maximal flow between $\underline{j}$ and $\underline{i}$ for an undirected network.)

Gomory and Hu [3] have shown that there are not more than n-1 different maximal flows and they can be found by solving only n-1 maximum flow problems (some of which are determined in smaller networks than the original).

To describe the Gomory and Hu approach, we introduce the following additional definition:

A connected set of arcs is one in which there is a path between every two nodes.

A *cycle* is a path in which the first and last node coincide.

A *tree* is a connected set of arcs with no cycles.

A *spanning tree* of a network is a tree that includes all the nodes of the network.

A *maximal spanning tree* is one in which the sum of the capacities of the arcs is maximal.

We now show that there are at most n-1 different values of f_{ij}, $(i,j\epsilon N)$.

Theorem (2): A necessary and sufficient condition for all f_{ij}, f_{jk}, f_{ik} to be the maximum flows between nodes $\underline{i}$ and $\underline{j}$; $\underline{j}$ and $\underline{k}$; and $\underline{i}$ and $\underline{k}$ respectively is that

(1) $$f_{ik} \geq \text{Min}\ (f_{ij},\ f_{jk})\ .$$

To show necessity: Suppose $f_{ik} < \text{Min}\ (f_{ij},\ f_{jk})$, then by the maximal flow-minimal cut theorem there is a cut $(X, \bar{X})$ with node $\underline{i}$ in X and $\underline{k}$ in $\bar{X}$, such that the capacity of the cut is equal to f_{ik} . Now node $\underline{j}$ belongs either to X or to $\bar{X}$.

If it is in X , $f_{jk} > C(X, \bar{X})$.

It it is in $\bar{X}$, $f_{ij} > C(X, \bar{X})$.

In both cases a contradiction is obtained.

(2) By induction $f_{ip} \geq \text{Min}\ (f_{ij},\ f_{jk},\ f_{kl},\ \ldots,\ f_{op})$.

To show sufficiency: Let H a "modified" network composed of the same nodes G but with arcs connecting every pair of nodes i, j and with capacities equal to the f_{ij}'s of G .

Now we construct the maximal spanning tree of H . We shall refer to the arcs of this tree as "links." Any f^{*}_{ip} of H that does not appear as a link of the maximal spanning tree satisfies:

(3) $$f_{ip} \leq \text{Min}\ (f_{ij},\ f_{jk},\ \ldots,\ f_{op})$$

* f_{ip} will denote the maximal flow of the link and is equal to its capacity.

where $f_{ij}, f_{jk}, \ldots, f_{op}$ form the unique path $(X_{ij}, X_{jk}, \ldots, X_{op})$ connecting nodes $\underline{i}$ and $\underline{p}$ within the tree. For if the inequality did not hold, the smallest link of the path could be removed and the direct link f_{ip} substituted to form a tree with larger value.

From (2) and (3), we conclude:

$$(4) \qquad f_{ip} = \text{Min}\ (f_{ij}, \ldots, f_{op}) \ .$$

As there are only n-1 links in the tree, we have shown that there are at most n-1 different values of the maximal flows.

In order to construct the maximal spanning tree, we introduce the notion of a condensed network.

Suppose that a maximal flow problem f_{st} has been solved between the source $\underline{s}$ and the sink $\underline{t}$ by locating a minimal cut $(X, \bar{X})$ with $\underline{s}$ in X and $\underline{t}$ in $\bar{X}$. To find f_{ij} where nodes $\underline{i}$ and $\underline{j}$ are at the same side of the cut $(X, \bar{X})$, say in X , we create a condensed network Q by shrinking all the nodes of $\bar{X}$ into a single node to which all the arcs of the minimal cut are attached. Several arcs joining the same pair of nodes can be replaced by a single arc whose capacity is given by the sum of the capacities of the arcs composing it.

Lemma:

The maximal flow between two ordinary nodes $\underline{i}$ and $\underline{j}$ in the condensed network Q is numerically equal to the flow f_{ij} in G .

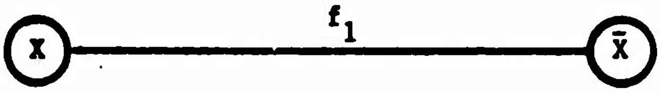
Proof:

f_{ij} in G is never greater than the flow between $\underline{i}$ and $\underline{j}$ in Q because shrinking the nodes of $\bar{X}$ in G to a single node $\bar{X}$ in Q is equivalent to making all arcs joining the nodes of $\bar{X}$ of infinite capacity.

On the other hand f_{ij} in G is never smaller than the flow in Q because if the f_{ij} in Q has any flow. γ from a node in X to $\bar{X}$, there is a set of arcs from X to $\bar{X}$ in the original network with a total capacity of at least γ. This statement can be made about every arc connecting the set X with the node $\bar{X}$.

The Gomory and Hu procedure for constructing the tree is the following:

Take any two nodes and find a minimal cut $(X, \bar{X})$ separating them. We represent this by two nodes



connected by a link with the value of the cut. In X are listed all the nodes of the set X and in $\bar{X}$ the others. We now repeat the process. Choose two nodes in X (or two in $\bar{X}$) and solve the flow problem in the condensed network in which $\bar{X}$ (or X) is a single node. The resulting cut has a value f_2 and is represented by a link connecting the two parts into which X is divided by the cut, say, X_1, and X_2. $\bar{X}$ is attached to X_1 if it is in the same part of the cut as X_1, to X_2 if it is in the same part as X_2.

The procedure is continued in the same way until each condensed node consists of only one node.

Examples of this procedure are given in subsequent illustrations of the algorithms of this paper, which use the Gomory and Hu method as a subroutine.

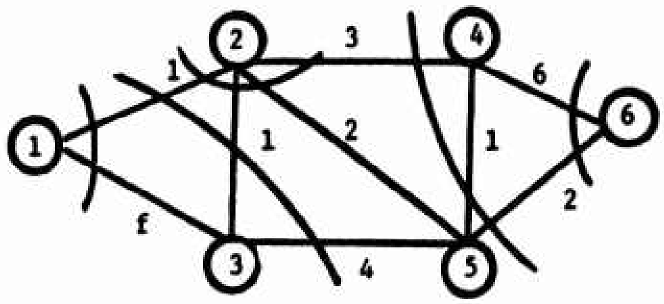
Lemma:

The maximal flow between any two nodes of the original network is simply the min $(f_{i1}, f_{i2}, \ldots, f_{ir})$ of the series of links of the tree connecting the two nodes.

We now prove in an indirect way that the tree constructed as above corresponds to the maximal spanning tree of the modified network.

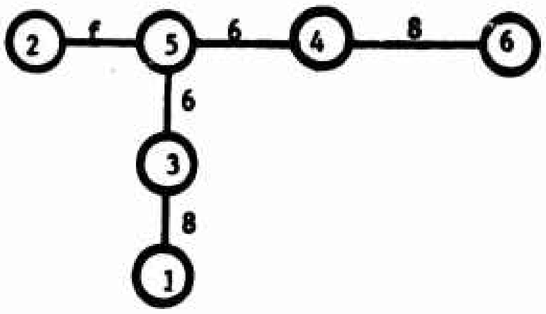
We have $f_{ij} < \min(f_1, f_2, \ldots, f_r)$; for each f_i on the path connecting nodes $\underline{i}$ and $\underline{j}$ is attached to a cut separating $\underline{i}$ and $\underline{j}$.

The following figures illustrate odr discussion.



numbers in arcs are capacities and
curve lines are the cuts

corresponding final tree

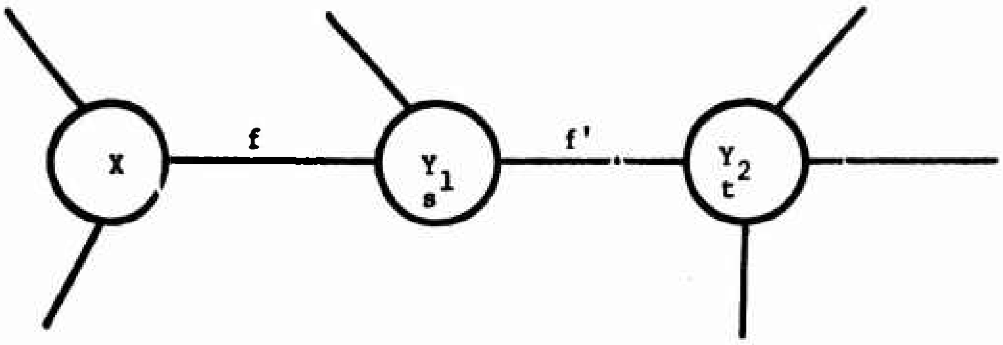


The proof of the reverse inequality is more difficult. It will be shown that, at any stage of the construction, if a link of capacity $\underline{f}$ joins nodes X and Y in the tree, then there is an x in X and a y in Y such that $f_{xy} = \underline{f}$.

The proof is by induction. It is clearly true at the first stage. Let's suppose it is true in the r^{th} stage and we will see that is still true in the next one. Consider X and Y , two contiguous condensed nodes of the tree at the r^{th} stage (r links, r+1 condensed nodes) and $x \in X$ and $y \in Y$ and $f_{xy} = \underline{f}$ = capacity of the link (X, Y) . Suppose Y has more than 1 node, at least has nodes $\underline{s}$ and $\underline{t}$, where the possibility than any .

one of this being equal to $\underline{y}$ is not excluded. In the next stage the set Y divides in Y_1 and Y_2 with $\underline{s}$ in Y_1 and $\underline{t}$ in Y_2 . We may assume that x is attached to Y_1 .

Now $\underline{s}$ and $\underline{t}$ are the two nodes such that $f_{st} = f'$ is the capacity of the link (Y_1,Y_2) . With respect to $\underline{f}$, there are two cases. If $\underline{y}$ is in Y_1, f_{xy} is still equal to $\underline{f}$. If $\underline{y}$ is in Y_2 , we shall show that $f_{xs} = \underline{f}$.



By Lemma 1, condensing Y_2 to a single node in the original network does not affect f_{xs} . Denoting by bars the maximal flows in the condensed network, we have:

$$\bar{f}_{xs} = f_{xs}$$

$$\bar{f}_{xy} > f_{xy} = f$$

$$\bar{f}_{yt} = \infty$$

$$\bar{f}_{ts} \geq f_{ts} = f' .$$

By Theorem 2, $\bar{f}_{xs} > \text{Min}\ [f_{xy}, \bar{f}_{yt}, \bar{f}_{ts}]$ and hence $f_{xs} = \bar{f}_{xs} \geq \text{Min}\ [f,f']$. But $f' \geq f$ because a cut of capacity f' separates $\underline{X}$ and $\underline{Y}$. Thus $f_{xs} = \underline{f}$, as was to be shown.

In consequence, the capacities of the links in the final tree actually represent maximal flows between its adjacent nodes.

Now from formulas 2 and 3 of Theorem 2, we get:

$$f_{ip} \geq \text{Min } (f_{ij}, f_{jk}, \ldots, f_{op})$$

$$f_{ip} \leq \text{Min } (f_{ij}, f_{jk}, \ldots, f_{op})$$

where ij, jk, ..., op , are the links of the tree joining $\underline{i}$ and $\underline{p}$, and we get $f_{ip} = \text{Min } (f_{ij}, \ldots, f_{op})$ showing that the final tree obtained above corresponds to the maximal spanning tree defined earlier. Because of the way it is obtained, we will call this tree the cut-tree.

In the next section, we discuss the influence on the cut-tree of a change in the capacity of 1 arc.

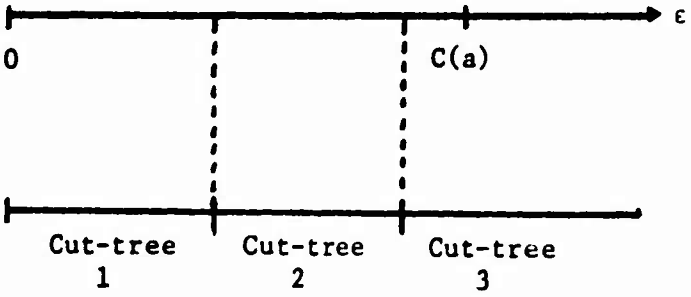
SECTION II

Sensitivity Analysis of Multiterminal Flow Networks Under Variation of the Capacity of 1 Arc

In this section an algorithm is developed for the effect on the maximal flow between any 2 nodes of the reduction or increase in the capacity of any arc. The capacity of all arcs except arc $\underline{a}$ (where arc $\underline{a}$ is any of the arcs of the network) will be considered constant, and arc $\underline{a}$ will have a capacity $C(a) - \underline{\varepsilon}$ where $\underline{\varepsilon}$ is a parameter that can take all the values in the range $[0 ; C(a)]$.

Our goal is to find the relation between the value of $\underline{\varepsilon}$ and the resulting cut-trees for an undirected network.

There are ranges of the value of $\underline{\varepsilon}$, let us say for all $\underline{\varepsilon}$'s between $\underline{\varepsilon_1}$ and $\underline{\varepsilon_2}$, for which one cut-tree is enough to describe the maximal flows between all pairs of nodes; moreover, there is a finite number of cut-trees that successively correspond to the variation of the value of $\underline{\varepsilon}$ from 0 to C(a) .



Algorithm

Step 1:

Construct the cut-tree of the network:

a) In each step of this construction put an initial capacity of zero to the arc a . Determine the maximal flow between the two nodes currently designated source and sink.

b) Reset the capacity of arc a to $C(a) - \varepsilon$ and continue maximizing the flow until it results in a cut-set. The value of the flow $f(X,\bar{X})$ through each cut-set $(X,\bar{X})$ can be an expression of one of these three types:

I) A constant α (this means that the flow didn't change from a) to b) and the cut-set is not through arc a) .

II) A function of ε as: $\gamma-\varepsilon$ where γ is a constant. (Cut-set through arc a .)

III) A constant β if $\varepsilon \leq k$. (The maximal flow did change from a) to b) and the cut-set is not through arc a for $\varepsilon \leq k$.) We record the value of k associated with each β .

Step 2:

a) If the cut-tree doesn't have any k it will represent the maximal flows between any pair of nodes, for $0 \leq \varepsilon \leq C(a^*)$ where $C^*(a)$ will represent the current value of $C(a)$, the value of the capacity in the specific cycle. (Initially $C^*(a) = C(a)$.)

b) If there are some values of k , take the smallest of them and recompute the maximal-flow between the two nodes of the link. Two things can happen: Either the value of such k remains the same in which case follow Step (3) or it is increased in which case it has to be compared to the others and if it is still the smallest, go to Step (3), otherwise go back to Step (2), phase b.

Step 3:

After all the computations in Step (2), we let $k^* =$ smallest {k recomputed}.

If $k^* \geq C^*(a)$, this is the last cycle and as in Step (2), the tree represents the maximal flows for all the positive values of the current capacity of arc <u>a</u> ; otherwise replace the capacity of arc <u>a</u> = $(C^*(a) - \varepsilon)$ by $((C^*(a) - k^*) + \varepsilon)$ where $(C^*(a) - k^*)$ will be the current value of $C^*(a)$ for the next cycle. Go to Step (1).

Note:

There is an example application of the algorithm at the end of the paper.

The network used is the same one used by Elmaghraby to facilitate a comparison of our method and his.

Justification of the Algorithm

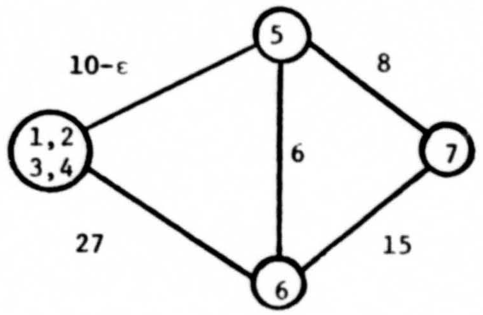
In the construction of the cut-tree, the maximum flow between two nodes i and j is determined in one of two ways: Either directly by a standard computation of the maximum flow between these nodes or indirectly as a result of computing maximal flows between other nodes. If the maximum flow is found by a direct computation, sending the maximum possible amount of flow through arcs other than arc <u>a</u> , and only subsequently using the arc <u>a</u> , we indeed get the maximum value of k for which the expression of the link represents the maximum flow between the two nodes.

If the maximum flow between nodes <u>i</u> and <u>j</u> is determined by computating of the flow between nodes <u>r</u> and <u>s</u> where <u>r</u> of <u>s</u> or both are different from <u>i</u> and <u>j</u> , and if the capacity of the link (i,j) is of the form $\beta(k)$, this value of k could have been constrained by any other link of the type II, and so a function of ε . In this case it will be detected in Step (2) of the algorithm.

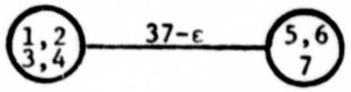
An example will help to understand this situation. The numbers in the trees and links are the respective capacities.

Let's suppose that after a first step in developing a cut-tree, we arrive at this situation:

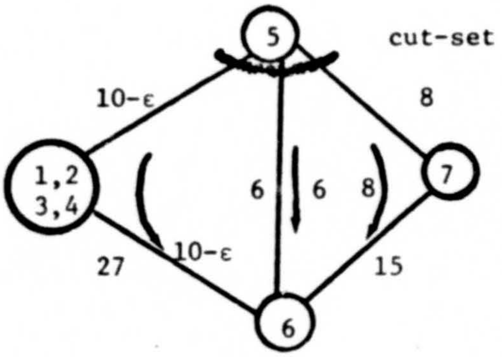
reduced network



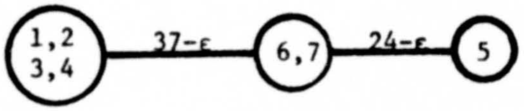
cut-tree



If we choose nodes ⑤ and ⑥ for the next step, we get: (The numbers in heavy black are the flows).

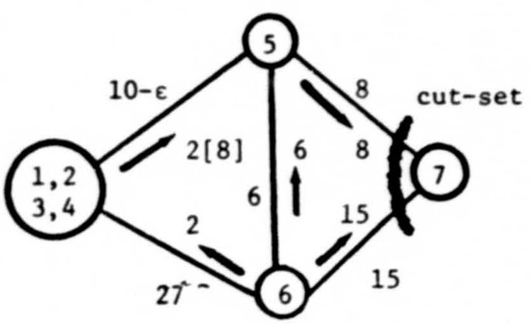


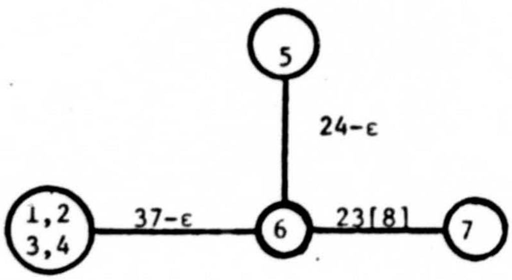
cut-tree



Maximum flow = 6 + 8 + (10 -ε) = 24 - ε .

Now solving for nodes ⑥ and ⑦





Maximum flow:

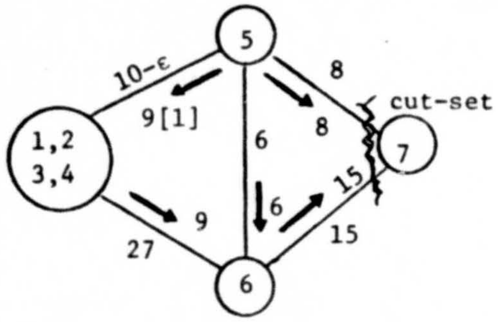
$$15 + 6 + 2\ [8] = 23\ [8]$$

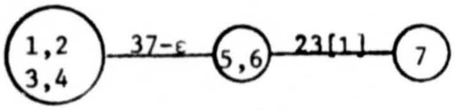
(That is, maximum flow is 23 if $\varepsilon \leq 8$.)

There is *only one link,* namely the one connecting nodes ⑥ and ⑦ that **has an associated** k **value:** k = 8 . This link was computed directly **because** *we did* solve the maximum flow between ⑥ and ⑦.

Now we will see what happens if that link is obtained in another way.

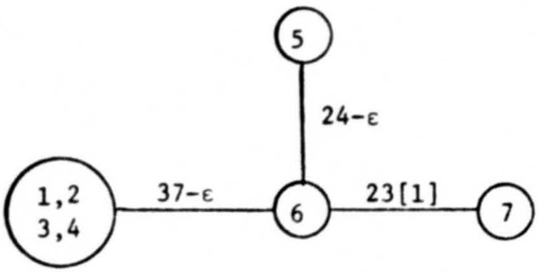
Beginning again, we now look for the maximum flow between ⑤ and ⑦.





Maximum flow: 8 + 6 + 9 [1] = 23 [1]

Between ⑤ and ⑥ we get the same maximum flow as before. So the cut-tree will now be:



We see that in this case we have $k = 1$, but the link ⑥ - ⑦ was not directly computed, (we computed links ⑤ - ⑦ and ⑤ - ⑥.) So we can compute ⑥ - ⑦ to check the value $k = 1$ and (as before) we get $k = 8$.

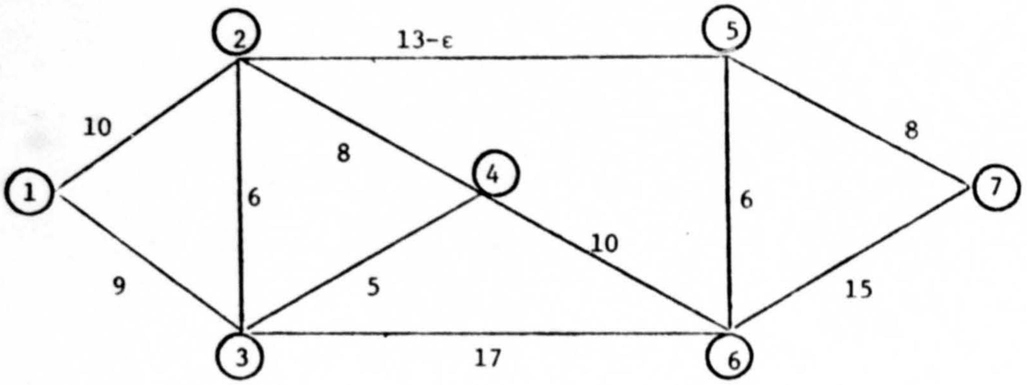
Applications

Following are some problems to which our algorithm can be applied.

1. Electricity line network of a country under partial failure or reparation of a line between two cities.
2. Similar problems with commodities like water or gas in a city, or flows of water for the event of fire in a "dry" city.
3. City of one-way (or almost one way) freeways where the direction of the freeways can be reversed, and where the people go to different places at specific times and the optimal number of lanes of a new highway is under study.
4. Constructing roads in a war problem "Jungle":
   Given a set of detachments (nodes) defending an area and very limited means of transportation (capacities), find a maximum flow of soldiers between the detachment that happens to be idle to the one under attack.

Example

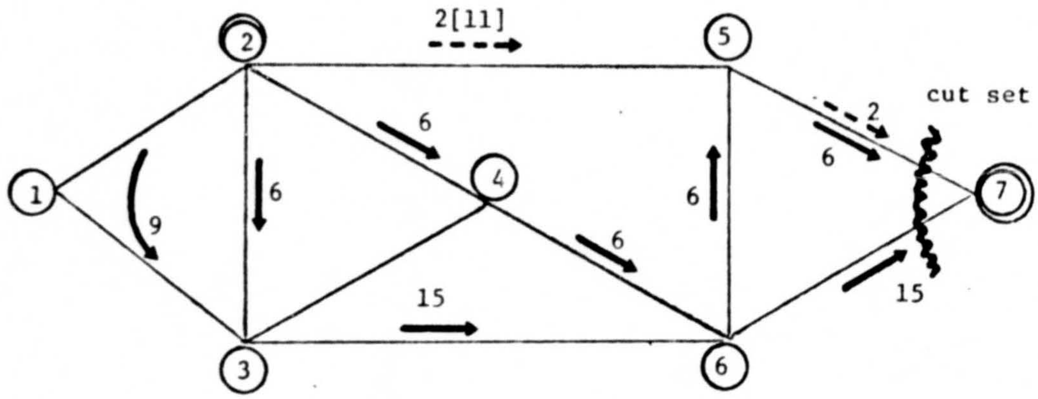
Original Network [†]



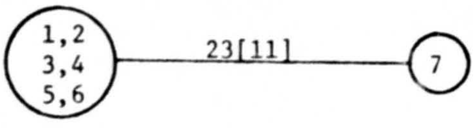
1st Cycle

Cut-set between ② and ⑦.

The maximum flow without using arc (2,5) was 21, using it, is 23.

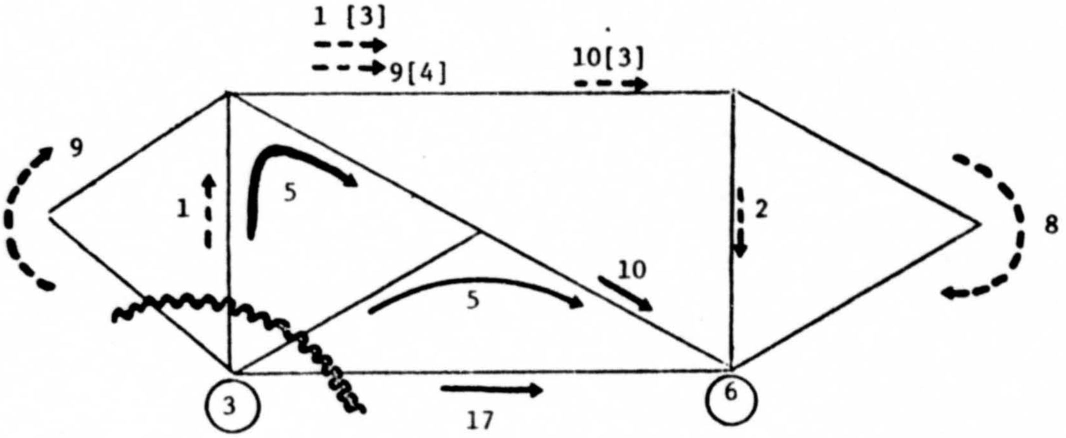


cut-tree

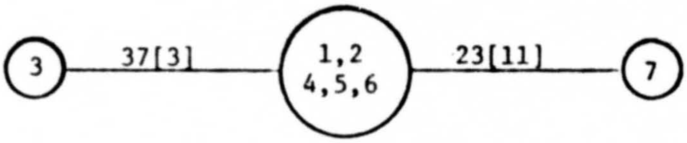


[†] Explanation: Black wavy lines are cut-sets, heavy black lines are flows when c(a) = 0 , dashes are using arc a .

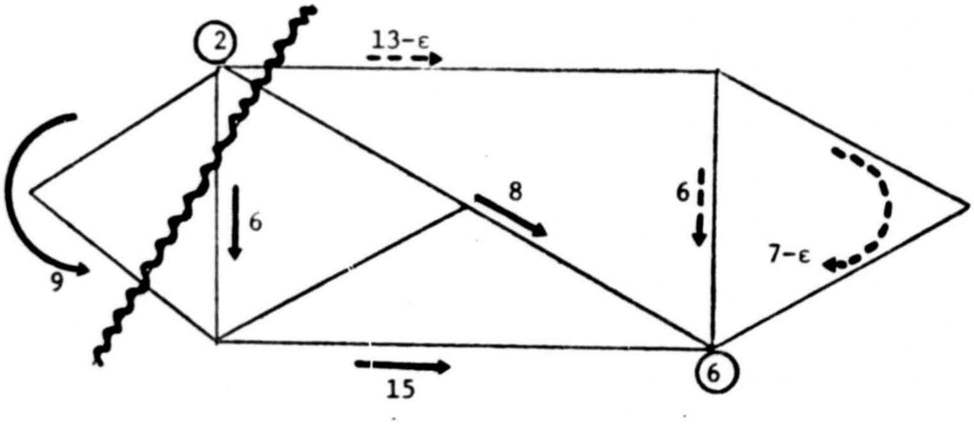
Cut set between (3) and (6).



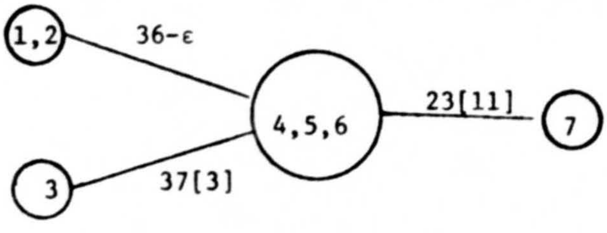
cut tree.



Cut set between (2) and (6).

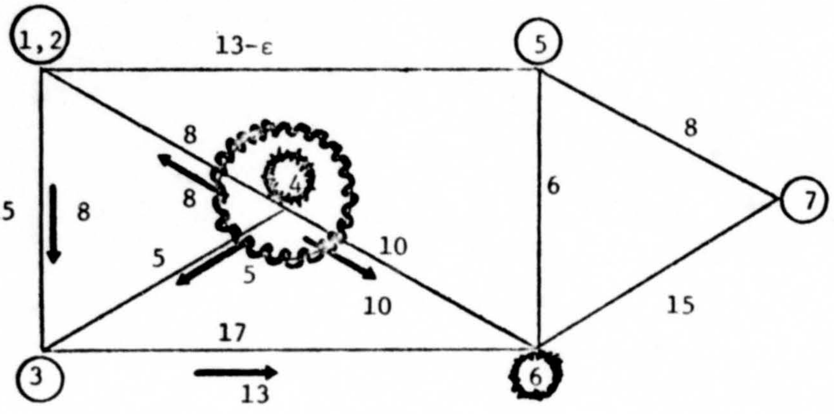


cut tree

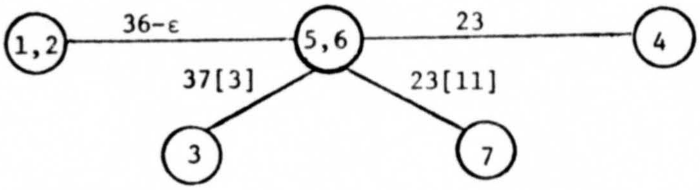


Cut set between ④ and ⑥.

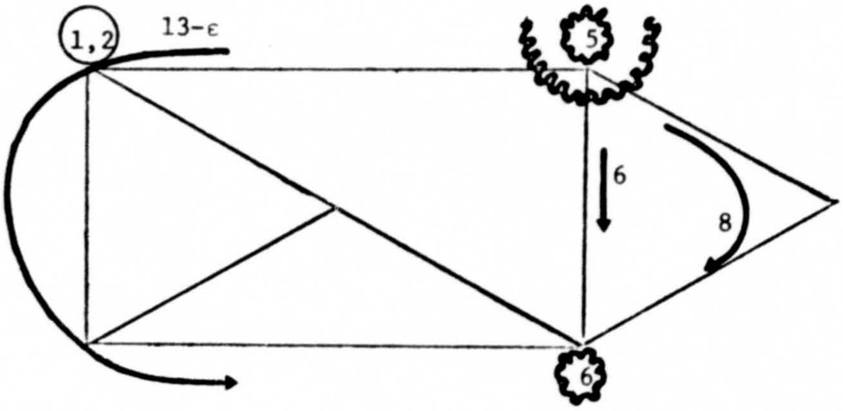
(reduced network)



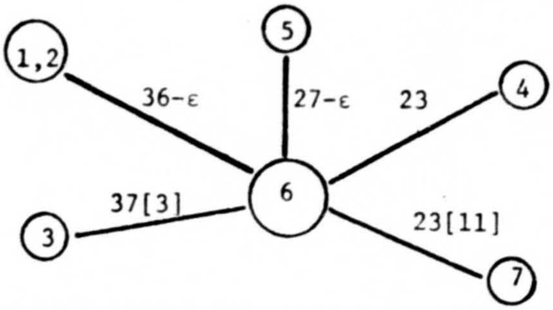
cut-tree.



Cut set between ⑤ and ⑥.



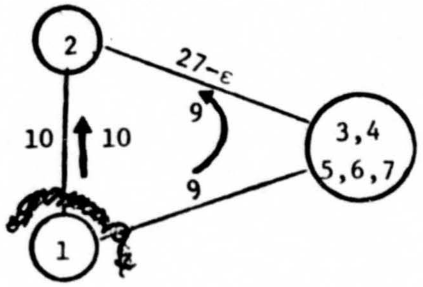
cut-tree.



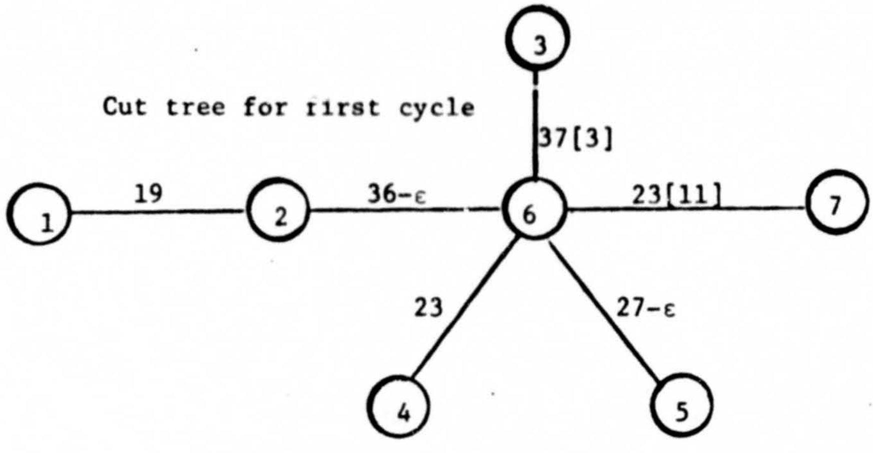
Cut set between ① and ② .

(reduced network)

(27-ε) does not interfere because maximum (ε) = 13



Cut tree for first cycle

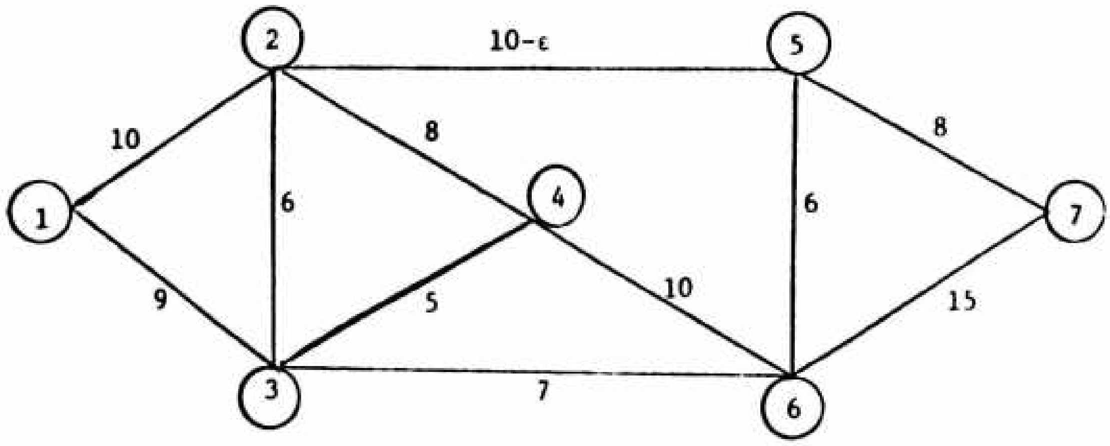


The link (3,6) with value 37[3] is the one that has the minimum k , k = 3 . As link (3,6) , was computed directly, it is not possible to improve that value and still have the same cut tree. This cut tree represents the maximum flows of the original network for $0 \leq \varepsilon \leq 3$.

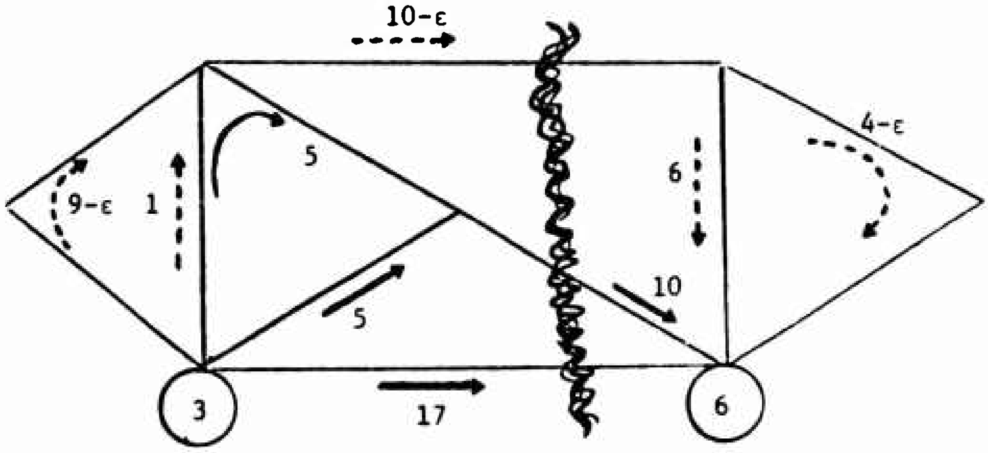
Now, we replace the capacity of arc (2,5) by 13-3-ε = 10-ε and begin again.

Second Cycle

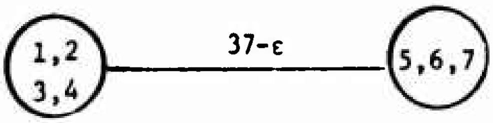
Original network with the capacity of arc (2,5) reduced to 10-ε .



Cut set between ③ and ⑥.

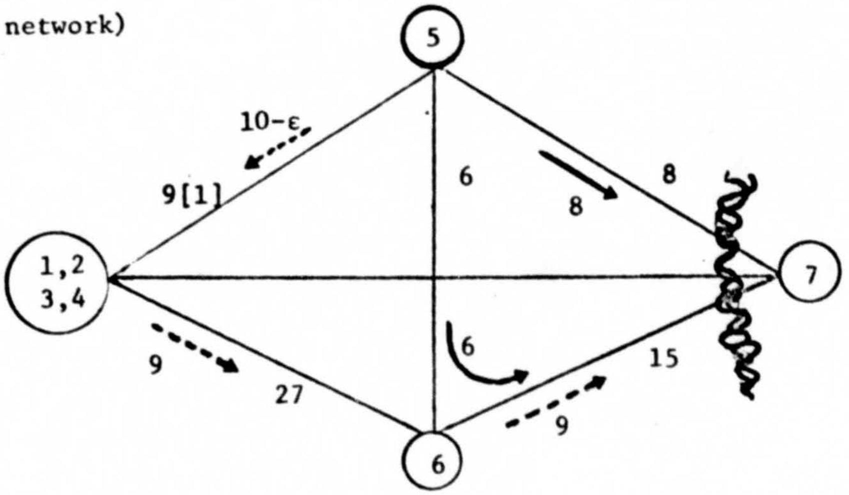


Cut tree .

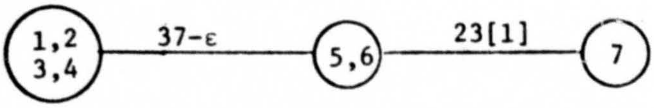


Cut set between ⑤ and ⑦.

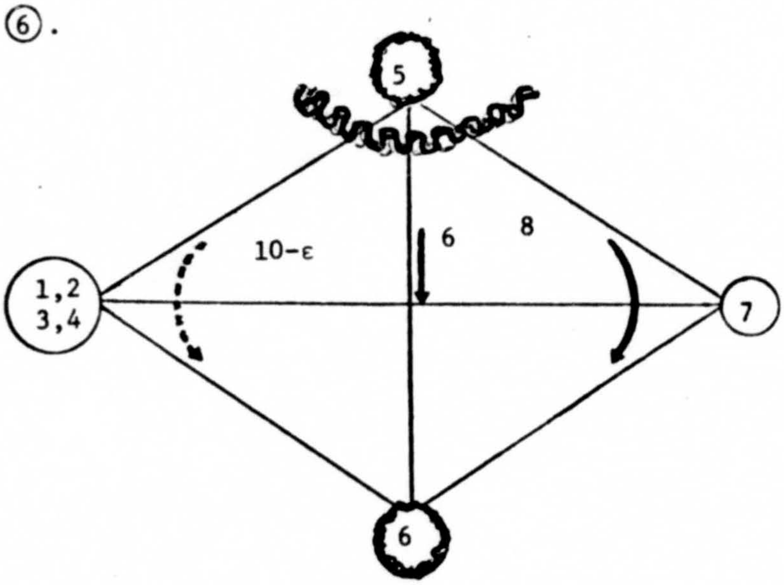
(reduced network)



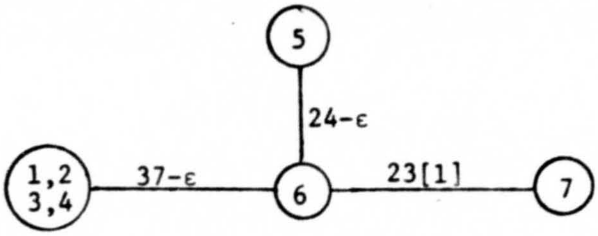
Cut tree.



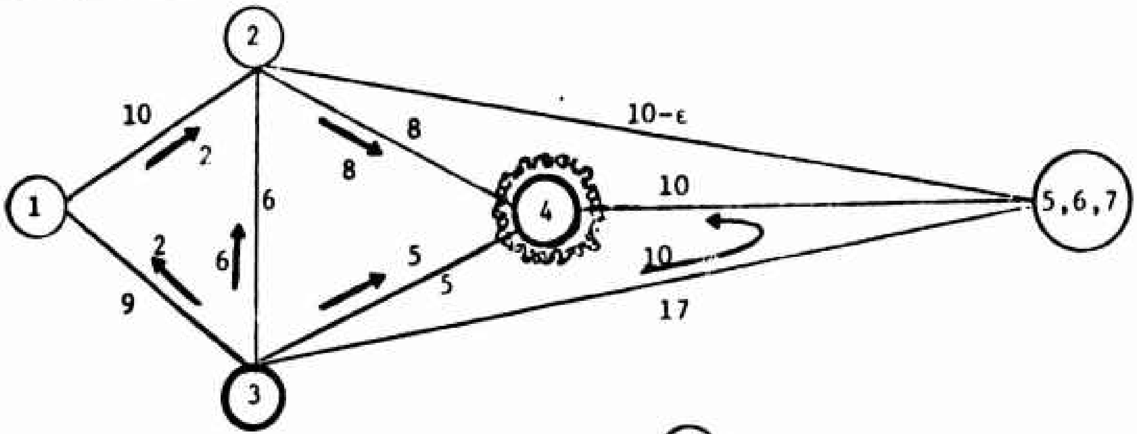
Cut set between ⑤ and ⑥.



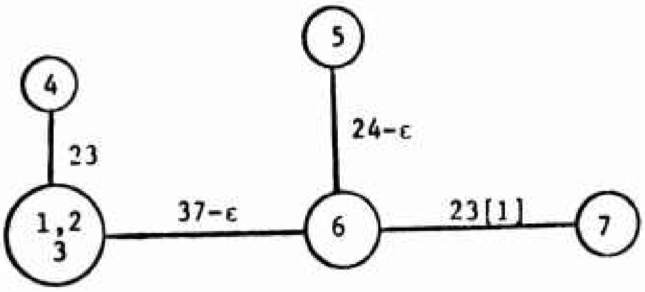
Cut tree.



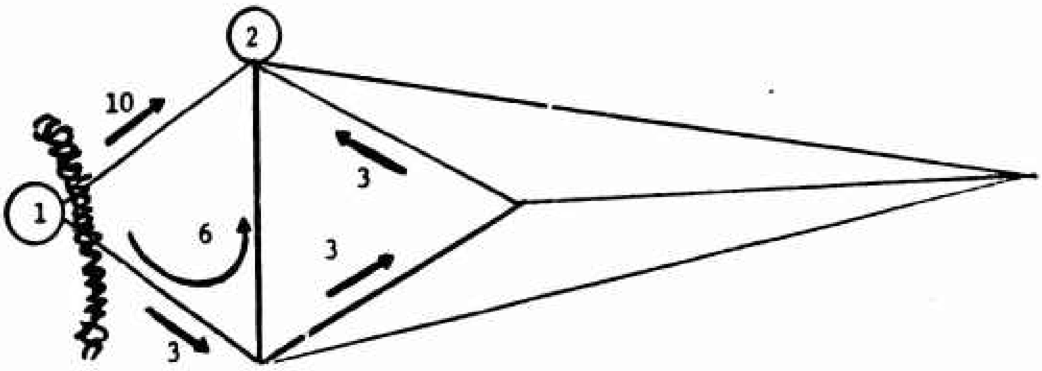
Cut set between ③ and ④.
(reduced network).



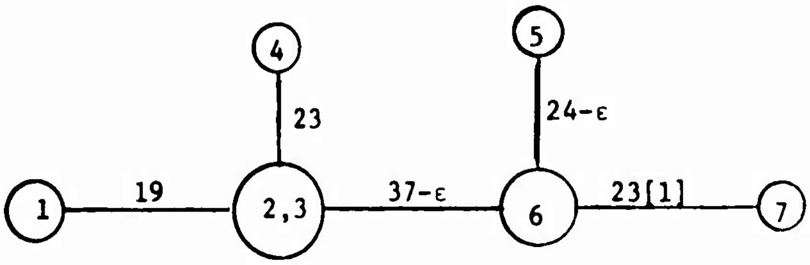
Cut tree.



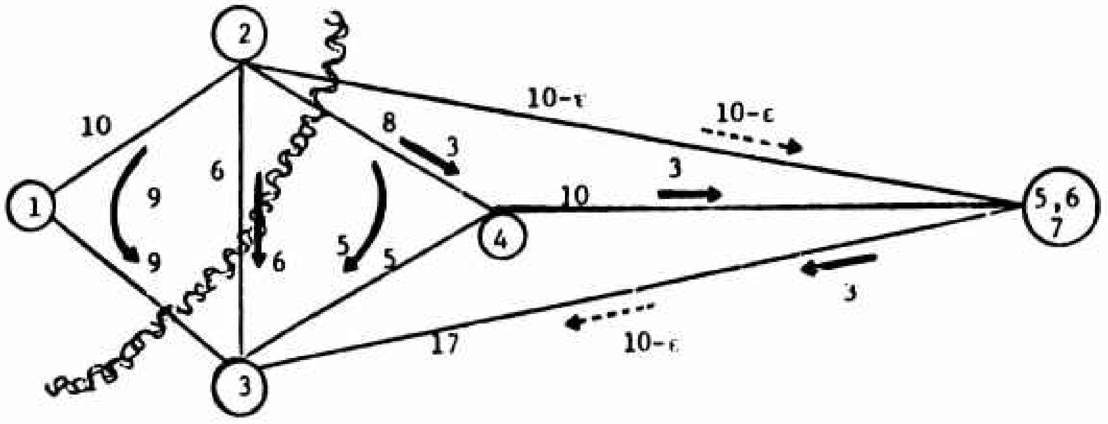
Cut set between ① and ②.



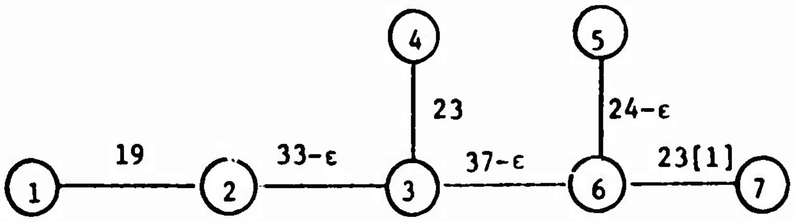
Cut tree.



Cut set between ② and ③.

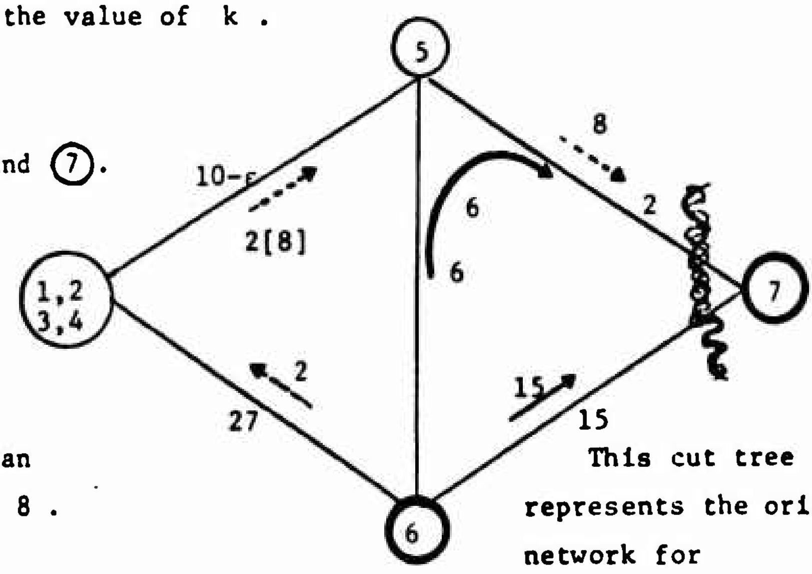


Cut tree.



The only link with k > 0 is link (6,7) with k = 1 ; as link (6,7) is not the result of the computation of the cut set between ⑥ and ⑦ but instead it was of ⑤ - ⑦ and ⑤ - ⑥, we have to compute the maximum flow between ⑥ and ⑦, to check the value of k .

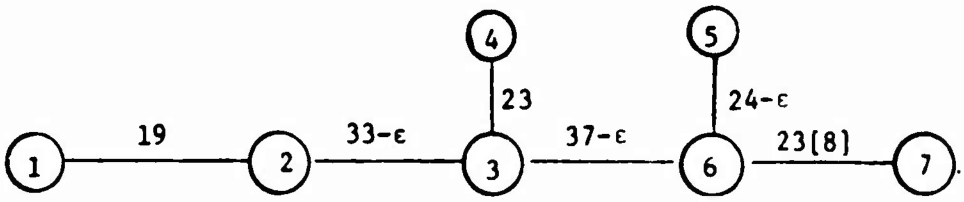
Cut set between ⑥ and ⑦.



Now we see that k can be improved till k = 8 .

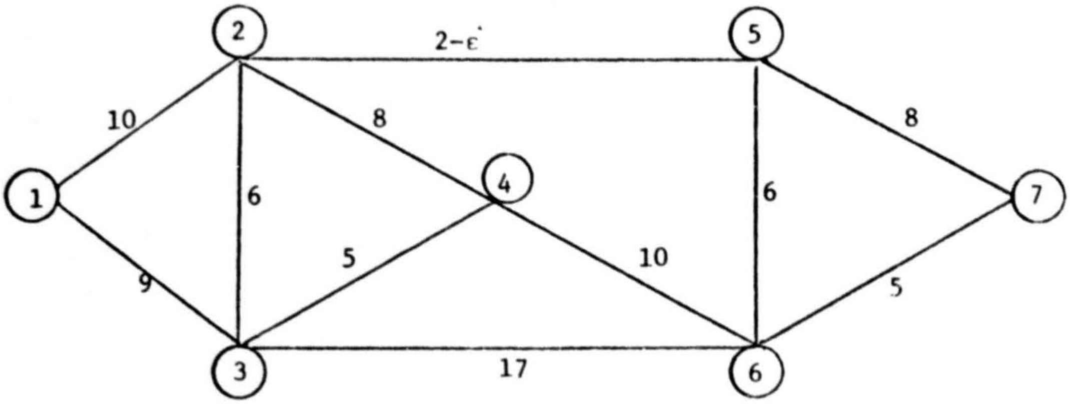
This cut tree represents the original network for $3 \leq \varepsilon \leq 8 + 3 = 11$

Cut tree.

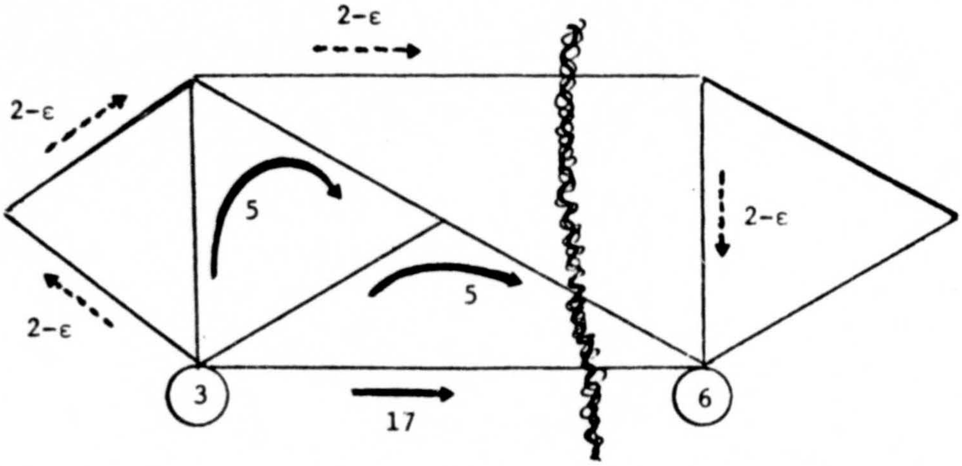


Third cycle

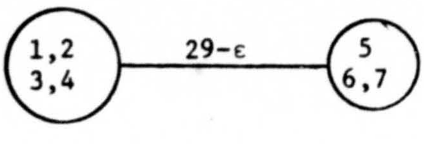
Original network with the capacity of arc (2,5) reduced to 2-ε .



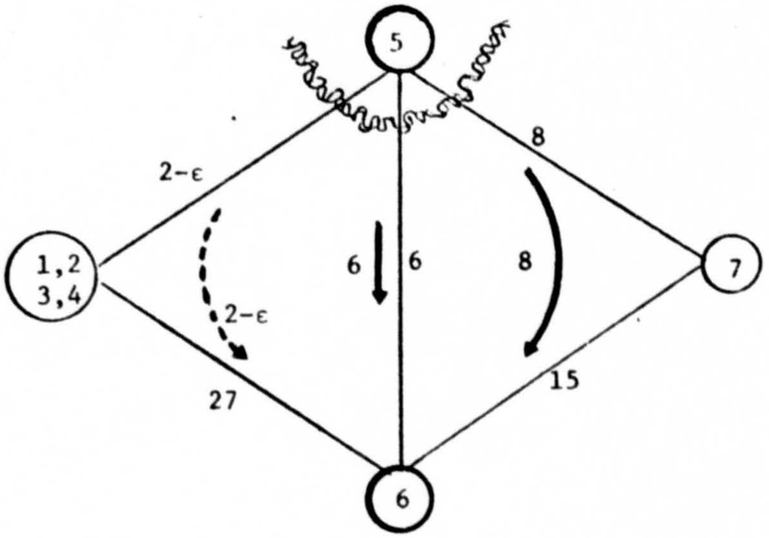
Cut set between ③ and ⑥.



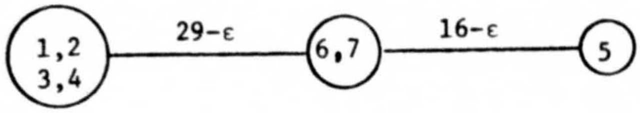
Cut tree.



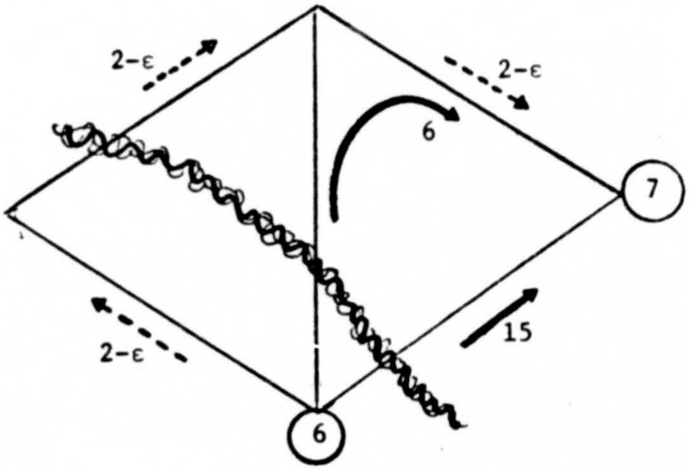
Cut set between ⑤ and ⑥.



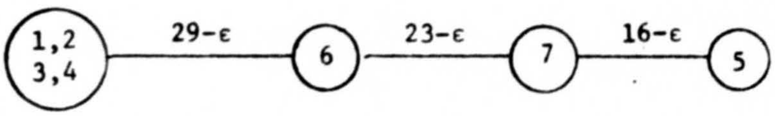
Cut tree.



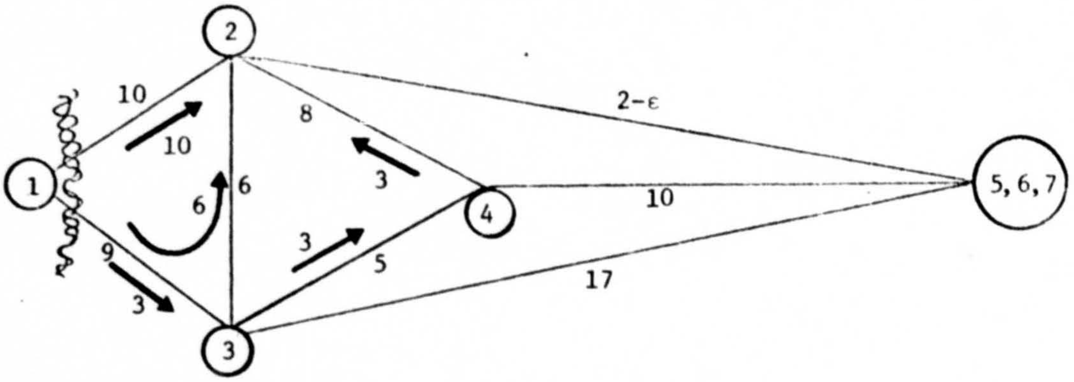
Cut set between ⑥ and ⑦.



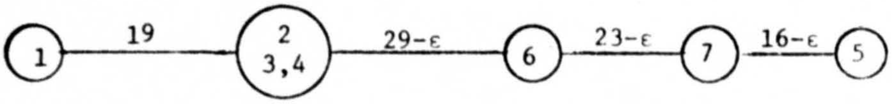
Cut tree.



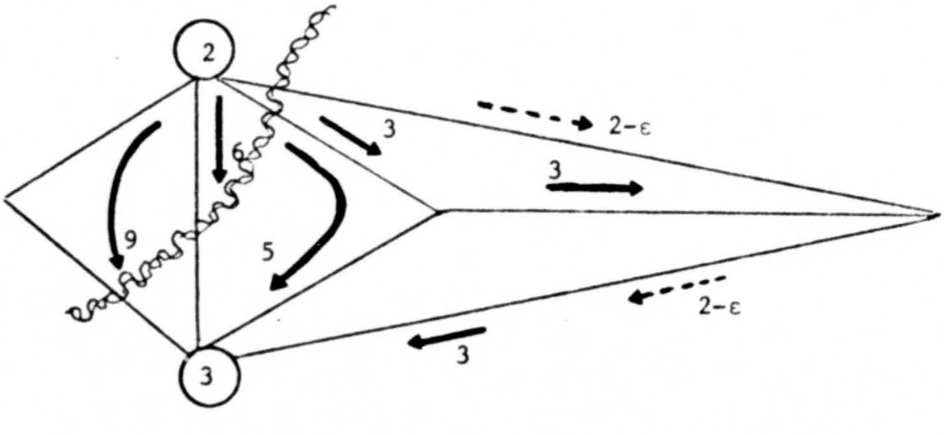
Cut set between ① and ②.



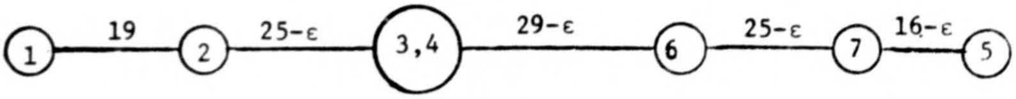
Cut tree.



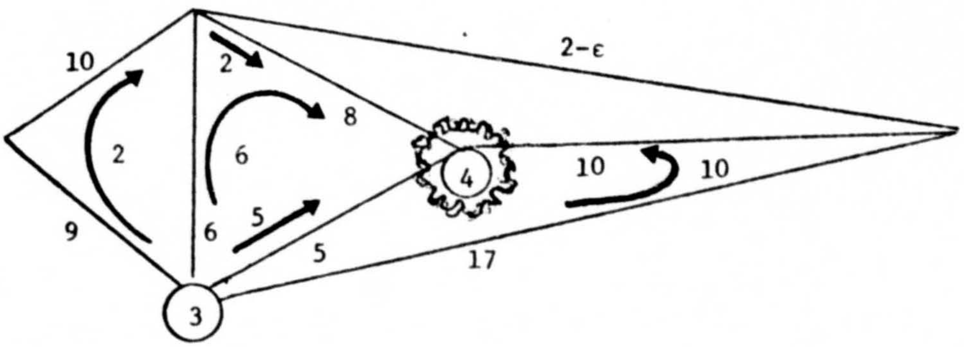
Cut set between ② and ③.



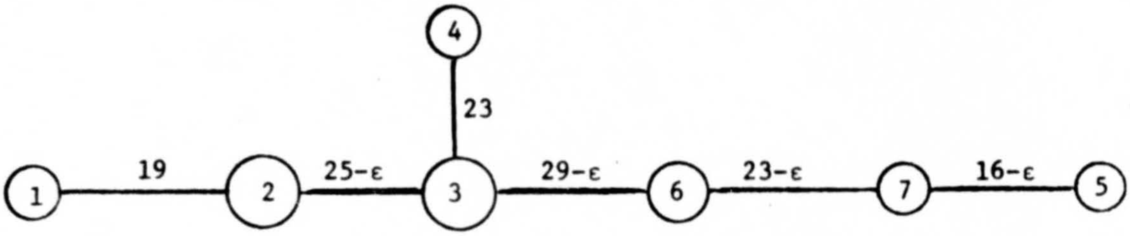
Cut tree.



Cut set between ③ and ④.



Definitive cut tree for third cycle.



We did not get any value of k , so, this cut tree represents the original networĸ for the maximal flows between any two nodes for $11 \leq \varepsilon \leq 13$. i.e., $\varepsilon = 11$ in the original network corresponds to $\varepsilon = 0$ in this cut treɛ; $\varepsilon = 13$ in the original network corresponds to $\varepsilon = 2$ in this cut tree.

i.e., maximal flow between ④ and ③ is 23 regardless of ε ,

" " ④ " " ② " the minimum of

(23 and 25−ε) .

" " ⑦ " " ① is the minimum of

(23−ε, 29−ε, 25−ε and 19) ,

etc.

SECTION III

SENSITIVITY ANALYSIS OF MULTITERMINAL FLOW NETWORKS UNDER VARIATION OF THE CAPACITY OF SEVERAL ARCS

We generalized the results of the previous section by allowing several of the arcs of the network to be linear functions of a parameter. Further generalizations (several parameters or nonlinear functions of them) will not be attempted in the present work.

The algorithm is very similar to the one discussed in the first part. Now we have arcs with capacities of three types:

$$C_{ij} \; ; \; C_{ij} + \alpha_{ij}x \; ; \; C_{ij} - \alpha_{ij}x \qquad\qquad \alpha_{ij} > 0 \; .$$

We will try to find the sequence of cut-trees that represent the maximal flows between any two arcs, when x varies over the real numbers between 0 and some arbitrary maximum x .

When the capacity of an arc becomes 0 , we simply erase such an arc not having defined negative capacities.

The Algorithm

Step 1:

Following the Gomory-Hu procedure, we construct the cut-tree of the network, in each step, using the labeling algorithm of Ford-Fulkerson to maximize the flow between nodes r and s (generic nodes).

As a result of the above, we will have in each arc (i,j) a flow, f_{ij} and a residual capacity r_{ij} representing the difference between the capacity and the flow through the arc. Also, we will have a cut-set and the value of its capacity.

f_{ij} can be of one of three types: f_{ij} ; $f_{ij} + \alpha_{ij}x$ and $f_{ij} - \alpha_{ij}x$

f_{ij} , $\alpha_{ij} \geq 0$.

r_{ij} can be of one of three types: r_{ij} ; $r_{ij} + \beta_{ij}$; $r_{ij} - \beta_{ij}x$.

r_{ij} , $\beta_{ij} \geq 0$.

For the flows and residual capacities of the types $f_{ij} - \alpha_{ij}x$ and $r_{ij} - \beta_{ij}x$, we compute the maximum value of x, call it k , that we can have before any flow or capacity becomes negative.

Three cases can occur:

1) There is no value of k , i.e., no flows go through arcs that can be affected by x . In this case, compute the next link.

2) The value of k is greater than the maximum x , we want to consider. Go to the next link.

3) The value of k is greater than 0 but smaller than maximum x . In this case, reset the value of the capacities that depend on x to : $C_{ij} + (k + \alpha_{ij})x$ and $C_{ij} - (k + \alpha_{ij})x$.

Compute again the same maximum flow in this network with modified capacities.

If both the cut-set and the expression of maximum flow don't change, we get a new value of k that should be added to the previous k , and if this value is still smaller than maximum x we repeat again and again [†] until either

[†] If the network is not too complicated an intelligent assignment of flows will frequently yield the optimal values in the first try without requiring successive adjustments. In the example that follows only two or three repetitions were required at each stage. The value of the residual capacity is written beside the flow only when necessary for understanding.

we are in Case 2. or either the cut-set or the expression of the maximum flow change in which case we have finished the computation of the link.

Step II:

Applies when we have completed the computation of the n-1 links. Now we have the cut-tree.

Take the link with the smallest value of k . If this value is larger than maximum x , we have finished, otherwise, if the link was computed directly go to Step 3; if it was not computed directly compute again the value of this link getting the same value of k or a larger one. Go back to Step 2.

Step III:

The value of k obtained in Step 2 gives the range for what the last tree is valid.

I.e., the first tree is valid for $0 \le x < k$, the second for $k_1 \le x < (k_1 + k_2)$ etc.

Reset the values of the capacities that are functions of x to

$$\left\{\begin{matrix} C_{ij} + k\alpha_{ij} + \alpha_{ij}x \\ \text{and} \\ C_{ij} - k\alpha_{ij} - \alpha_{ij}x \end{matrix}\right\} \quad \text{instead of} \quad \left\{\begin{matrix} C_{ij} + \alpha_{ij}x \\ \\ C_{ij} - \alpha_{ij}x \end{matrix}\right\}$$

go back to Step I for a new cycle.

Justification of the Algorithm

In Step I, we maximize the flow between two nodes. This can be accomplished in many ways. Each way leaves some residual capacities and flows as a function of x that tend to zero with the increase of the value of x , yielding an upper bound k on the value of x .

Changing the capacities of the arcs of the types

$$\left\{\begin{matrix} C_{ij} + \alpha_{ij}x \\ \\ C_{ij} - \alpha_{ij}x \end{matrix}\right\} \qquad \text{to} \qquad \left\{\begin{matrix} (C_{ij} + k\alpha_{ij}) + \alpha_{ij}x \\ \\ (C_{ij} - k\alpha_{ij}) - \alpha_{ij}x \end{matrix}\right\}$$

we can change the amount of flow carried by the different arcs to get the minimum restriction and hence the maximal value of k .

In Step II, as in the previous algorithm (and for the same reasons), we compute again the k corresponding to the link with minimum k , if it was not computed directly.

Important

In any direct computation the value of k can get larger or stay the same, but never get smaller, because it is limited by all the restrictions imposed by the cut-sets that were computed indirectly to get the link we are considering.

When we compute the link directly, fewer restrictions result and the maximum value of a function cannot diminish.

Examples of Applications

1) The network consists of one way highways, some of asphalt and some of concrete.

The capacity of the highways will be given the number of lanes, and the speed of the cars on the asphalt lanes decreases proportionally (for some range) to the amount of rain. Thus, the capacity of a four lane concrete highway is 4, while the capacity of a 4 lane asphalt highway is 4 - 4x , etc.

2) The network consists of electric lines going from power plants to the cities. Some of the plants are hydraulic plants and others are thermal. Because of increase of coal costs it is under consideration to decrease the capacity of the coal plants and to increase the capacity of the hydraulic ones. (In this case artificial arcs with the capacities of the plants have to be added to the network.)

Comments for the Second Algorithm

The problem solved by the second algorithm was taken from an unpublished paper by Elmaghraby. The example presented is also his, to facilitate an eventual comparison of methods.

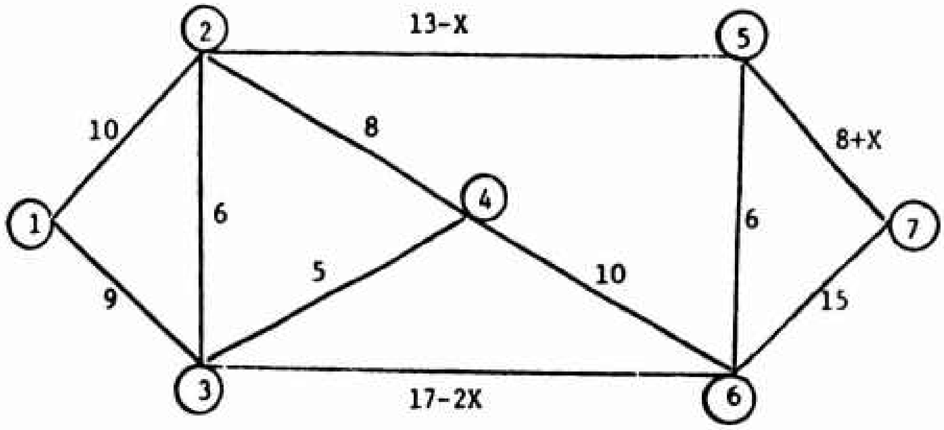
Step I of the present algorithm is essentially similar to the one in his paper.

Step II, which differs from the approach of Elmaghraby, provides a way to bypass the construction of superfluous cut-trees.

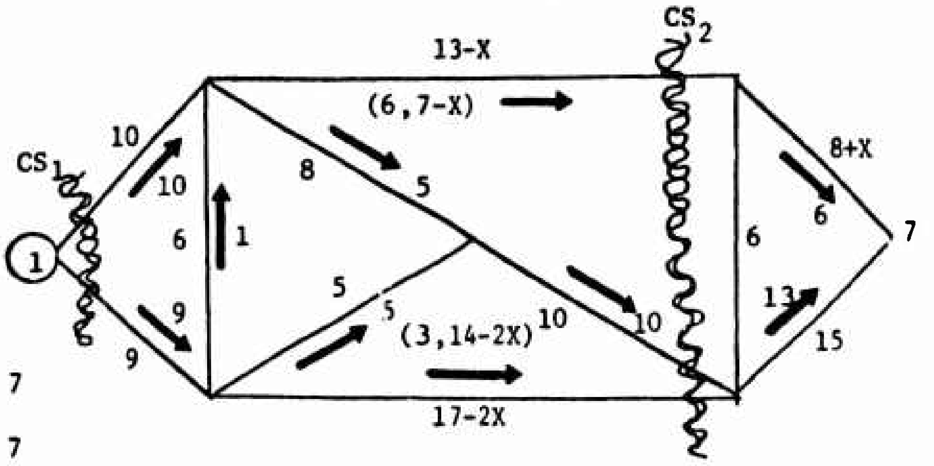
Example

Original network

Sensitivity analysis will be done for X in the range for this m lti-terminal network. $0 \leq X \leq 6.66$



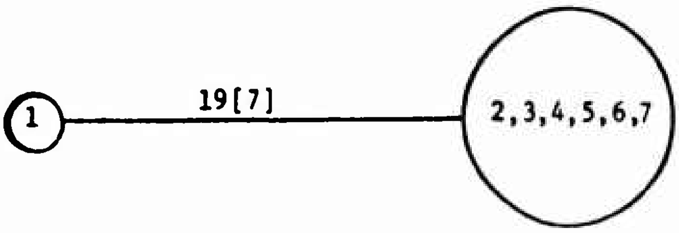
Construction of the first cut-tree or (first cycle).



$F - X = 0 \longrightarrow X = 7$

$14 - 2X = 0 \longrightarrow X = 7$

No iteration was done because it is evident that for $X > 7$ the cut-set changes.†



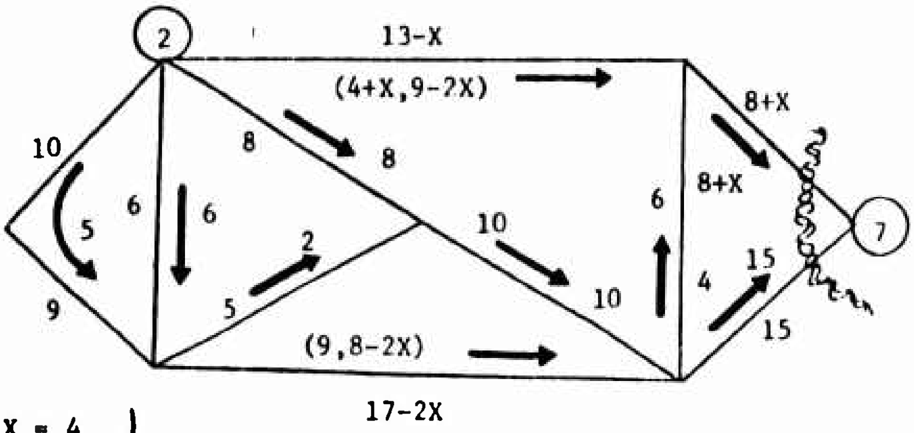
Flow = 19

CS_1 is the cut-set for $X \leq 7$.

CS_2 is the cut-set for $X > 7$.

† The same thing happens most of the time throughout this example.

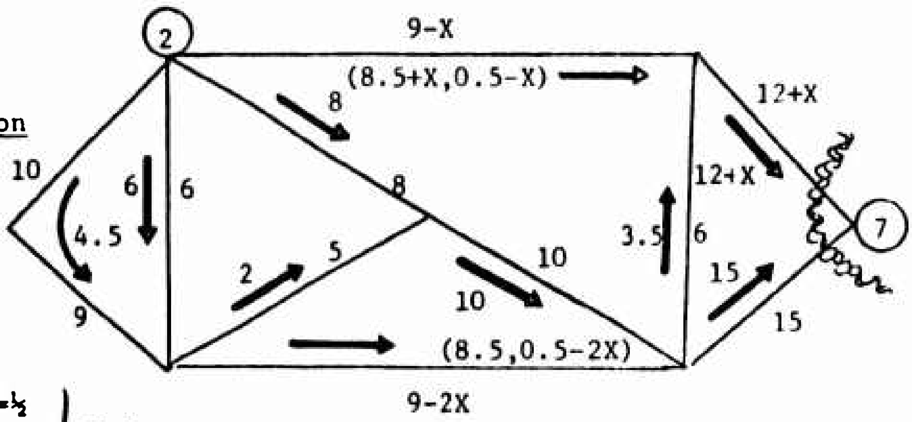
First iteration



$$\left.\begin{array}{l} 8 - X = 0 \longrightarrow X = 4 \\ 9 - 2X = 0 \longrightarrow X = 4.5 \end{array}\right\} X = 4$$

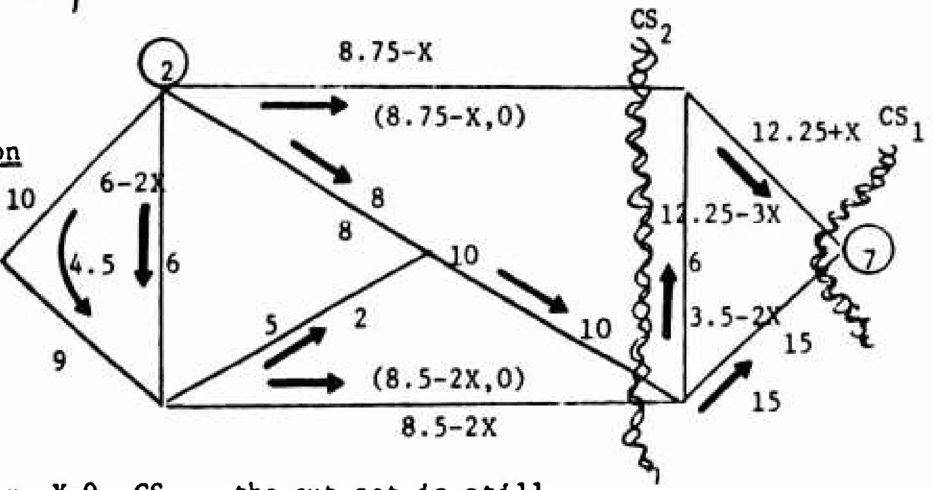
Flow = 23+X

Second iteration



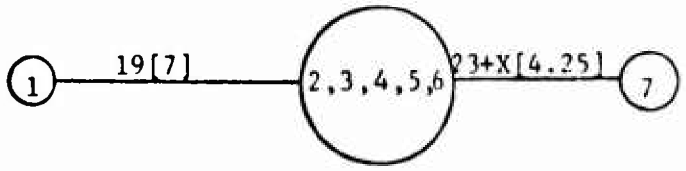
$$\left.\begin{array}{l} \tfrac{1}{2}-X=0 \longrightarrow X=\tfrac{1}{2} \\ \tfrac{1}{2}-2X=0 \longrightarrow X=\tfrac{1}{4} \end{array}\right\} X=\tfrac{1}{4}$$

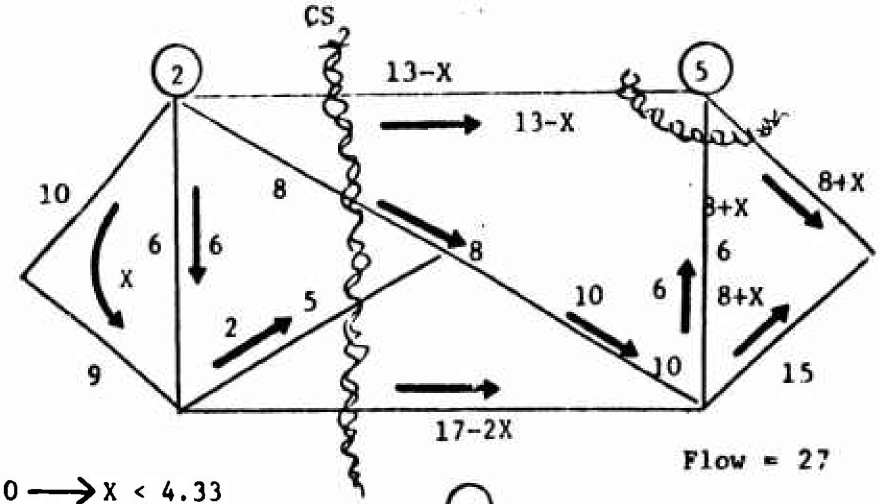
Third iteration



We see that for X=0 CS_1 , the cut-set is still the same, but for any positive value of X , is CS_2 .

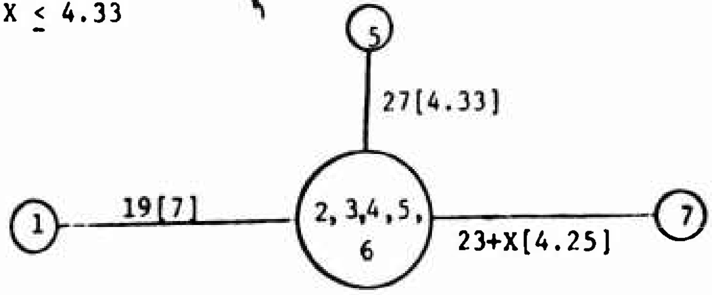
$k=4+\tfrac{1}{4}=4.25$

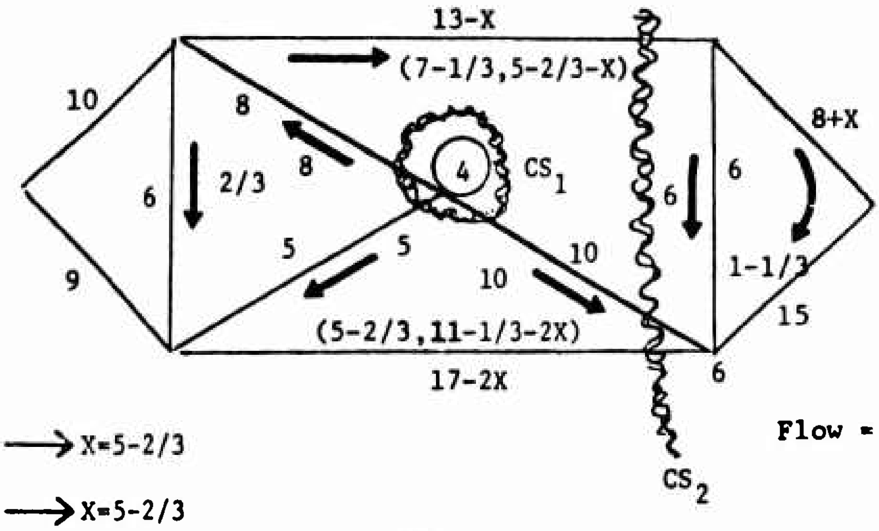




Flow = 27

13 - 3X = 0 ⟶ X ≤ 4.33

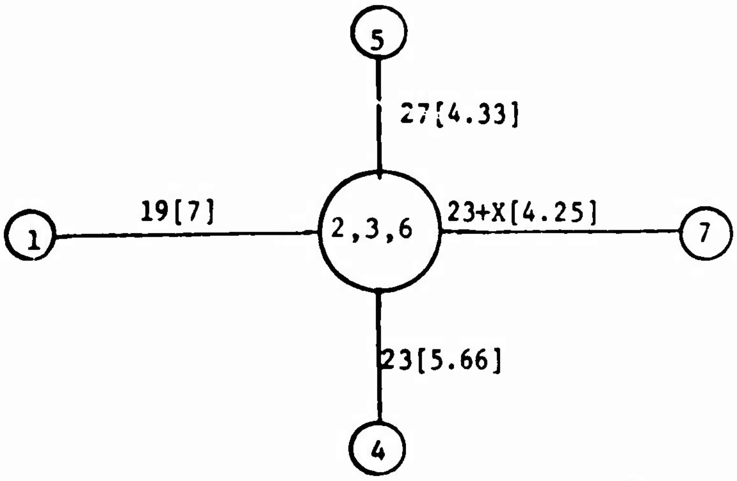




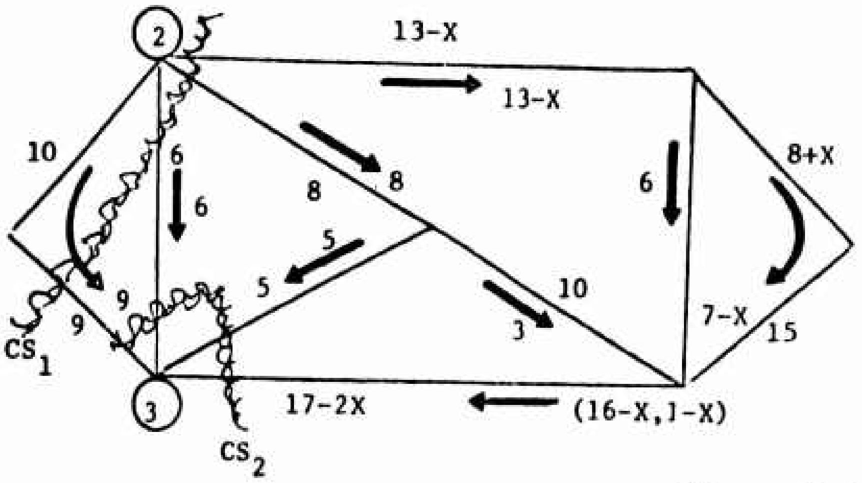
Flow = 23

11-1/3-2X=0 ⟶ X=5-2/3

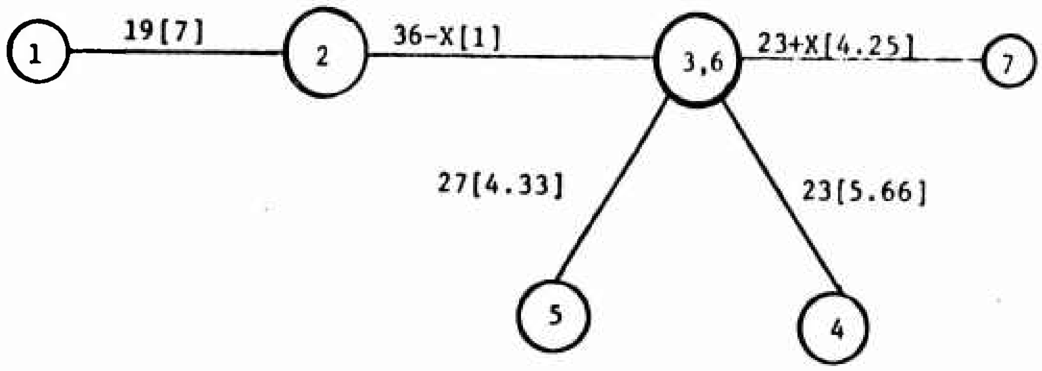
5-2/3-X=0 ⟶ X=5-2/3

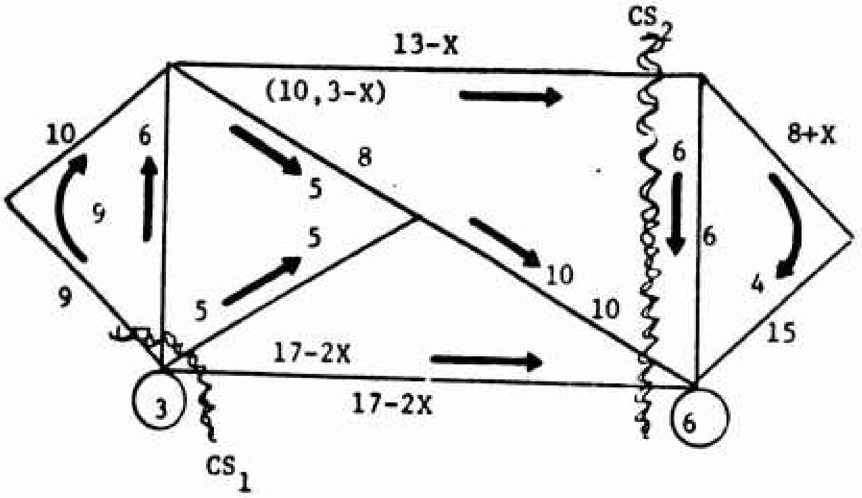


1-X=0 ⟶ X=1

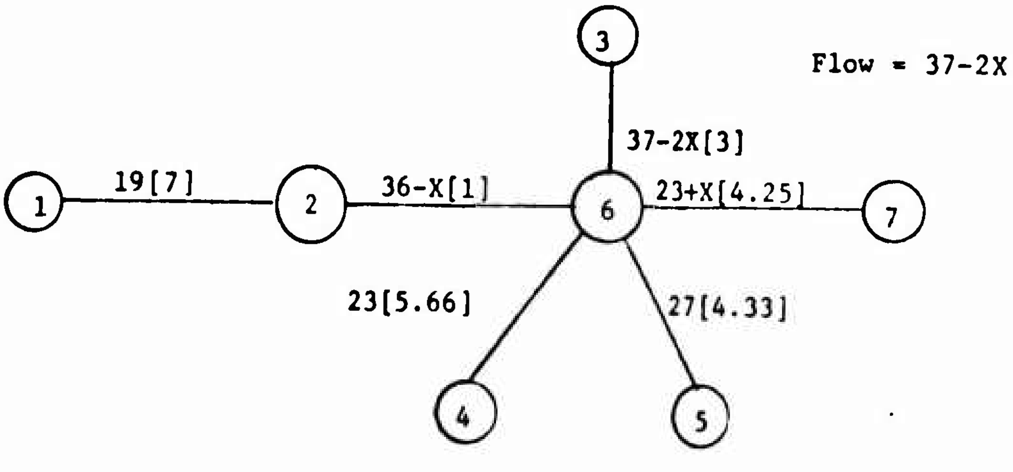


Flow = 36-X

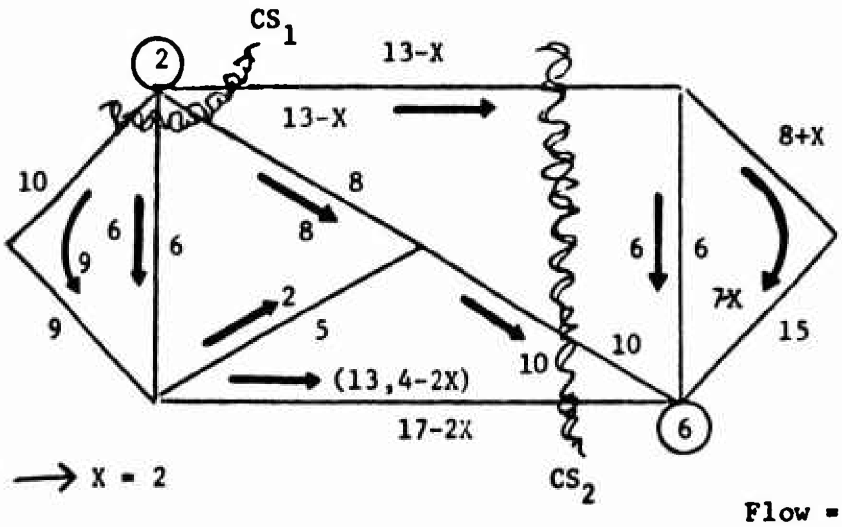




3-X=0 ⟶ X=3



Link (2,6) has the smallest value of k , $k = 1$. As this link was not computed directly but instead of it were (2,3) and (3,6), we have to check its value.



$4 - 2X = 0 \longrightarrow X = 2$

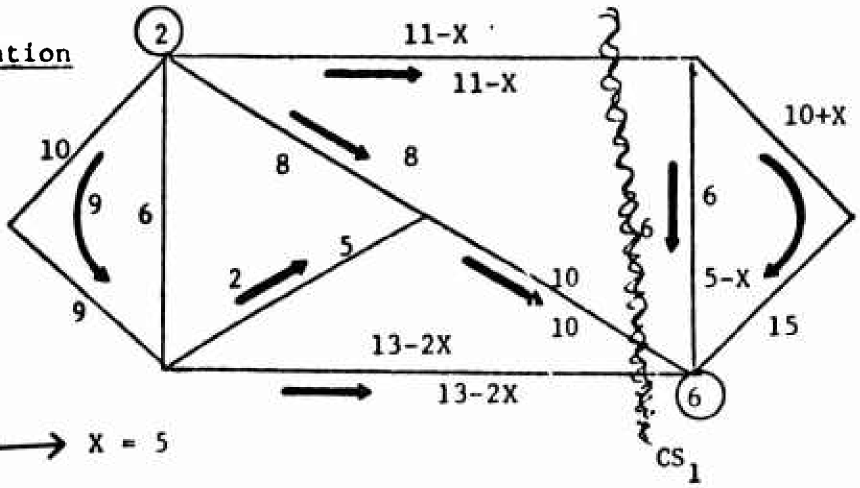
Flow = 36 - X [2]

As we see, the value of k , has been improved to $k = 2$. It happens that it is still the smallest, and this time it has been computed directly, so we cannot improve any more.

This cut-tree (the last one) represents the maximum flows in the network for values of x : $0 \leq X \leq 2$. The next cut-tree will represent the maximum flows for values of X larger than 2 . The upper limit will be found in the same way we got the upper limit 2 for the first cut-tree.

Second Cycle

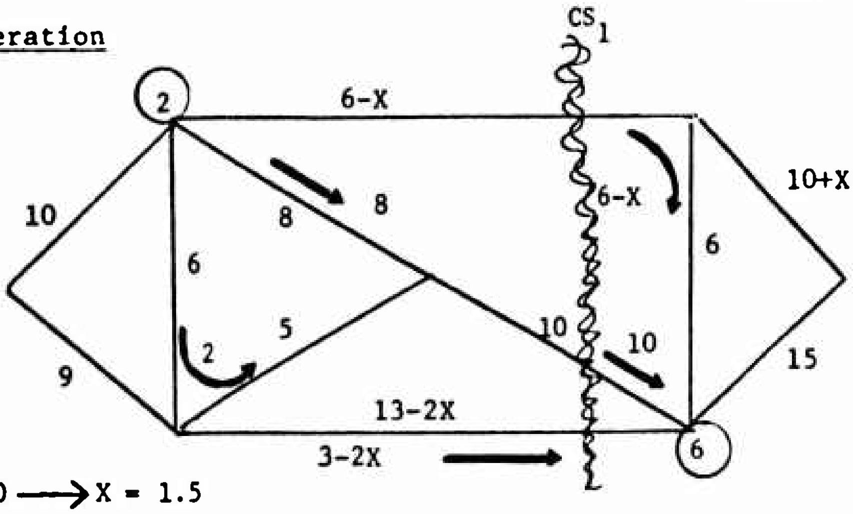
first iteration



$5 - X = 0 \longrightarrow X = 5$

Flow = 34 - 3X

second iteration

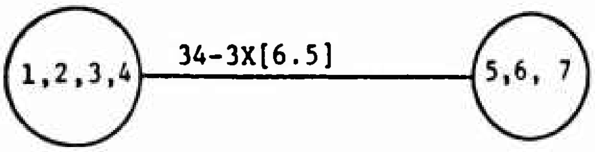


$3 - 2X = 0 \longrightarrow X = 1.5$

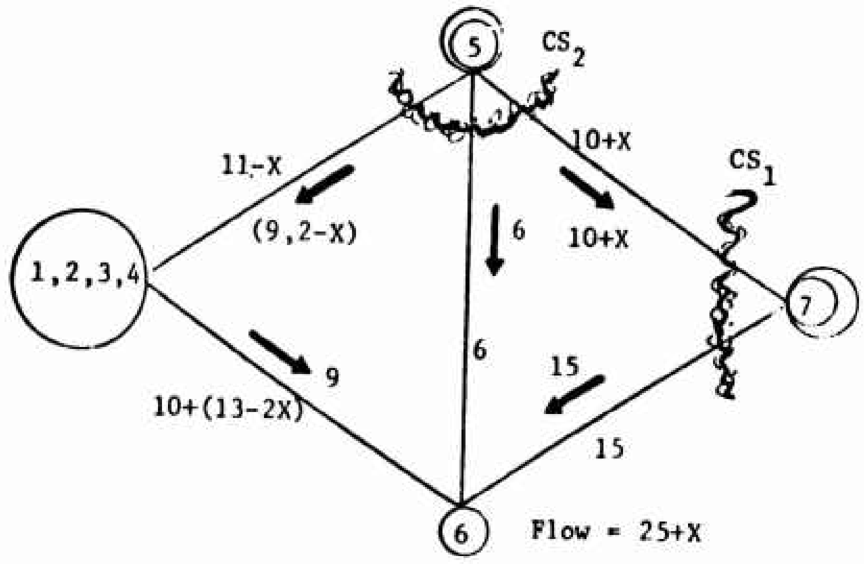
Flow = 19-3X=34-(3.5)-3X

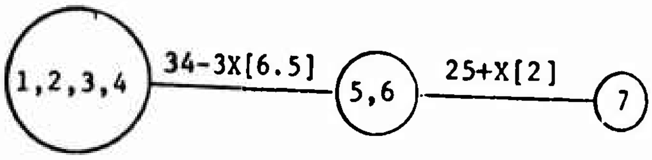
After that the cut-set is the same, but the structure of the link changes to 16 - 2X .

k = 5 + 1.5 = 6.5



$2 - X = 0 \longrightarrow X = 2$

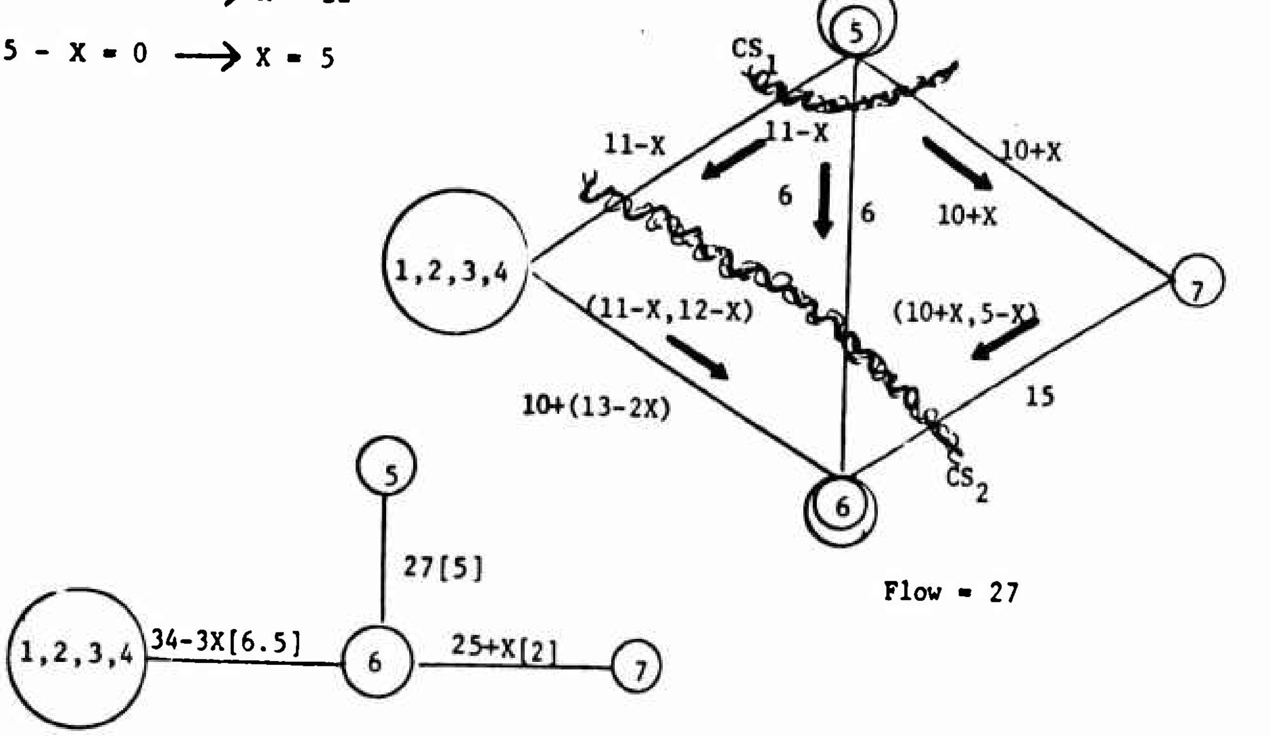


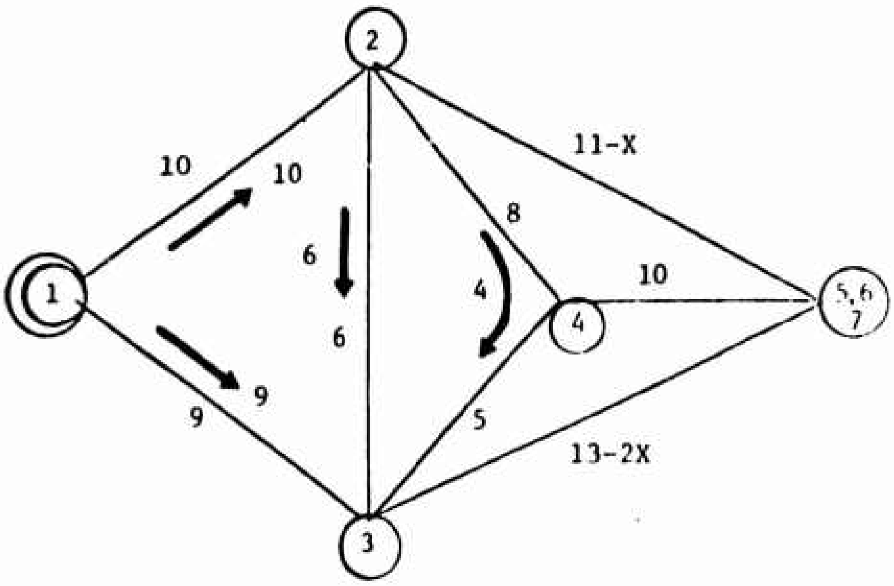


$11 - X = 0 \longrightarrow X = 11$

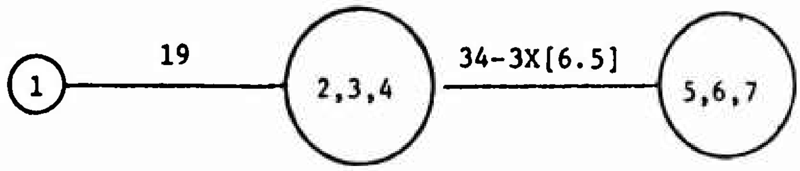
$12 - X = 0 \longrightarrow X = 12$

$5 - X = 0 \longrightarrow X = 5$





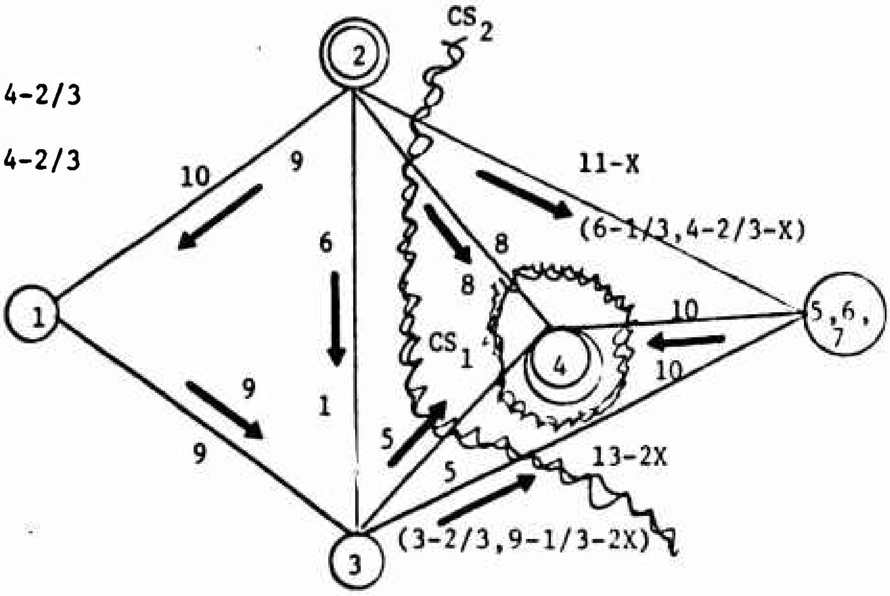
No restriction in X .



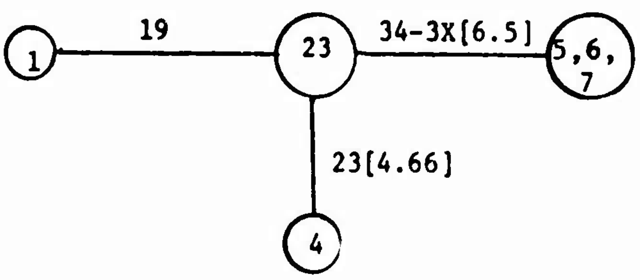
Flow = 19

$9\text{-}1/3 - 2X = 0 \longrightarrow X = 4\text{-}2/3$

$4\text{-}2/3 - X = 0 \longrightarrow X = 4\text{-}2/3$

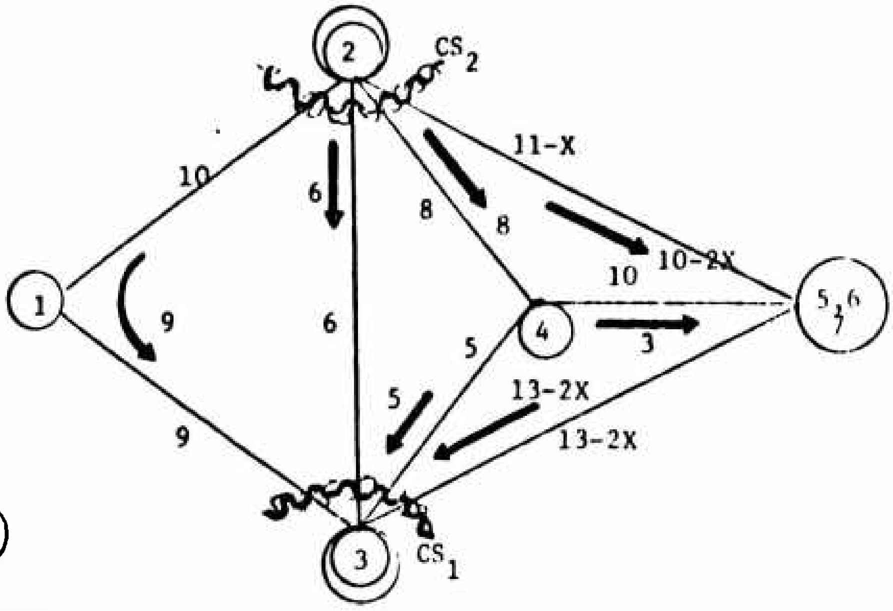


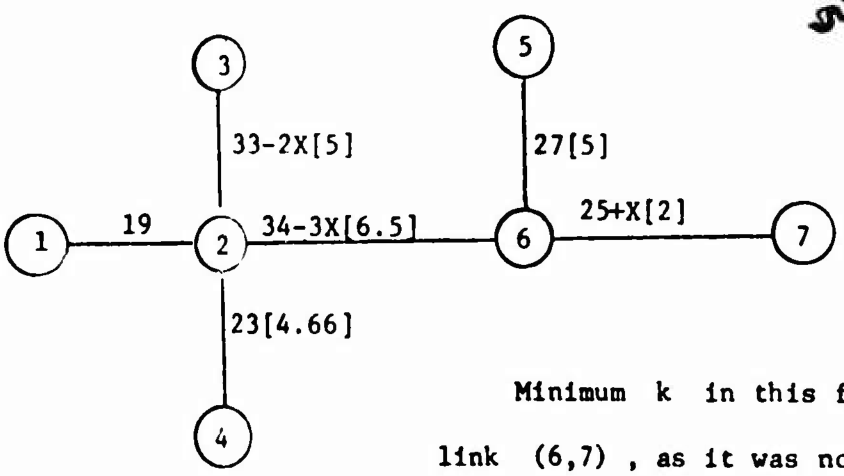
Flow = 23



13 - 2X = 0 ⟶ X = 6.5

10 - 2X = 0 ⟶ X = 5





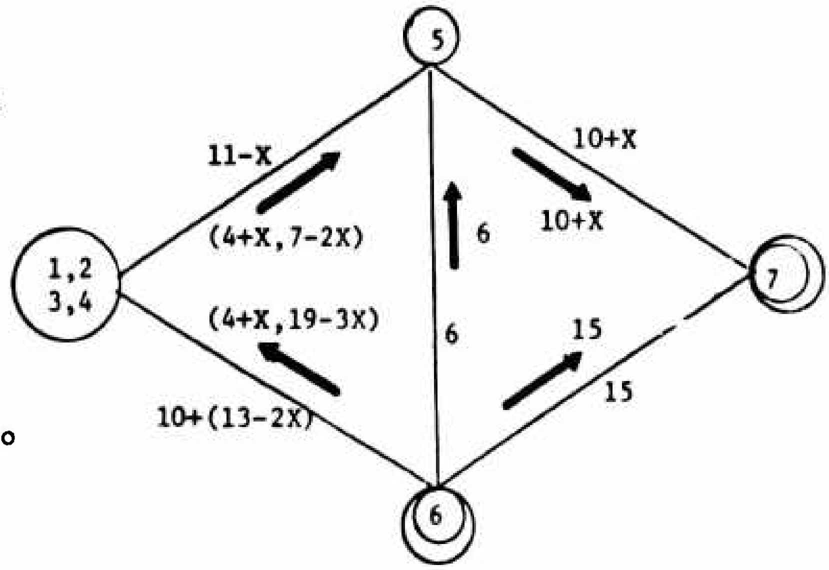
Flow = 33 - 2X

Minimum k in this final cut-tree is 2 , from link (6,7) , as it was not computed directly, it is necessary to compute it again.

19 - 3X = 0 ⟶ X = 6-1/3

7 - 2X = 0 ⟶ X = 3.5

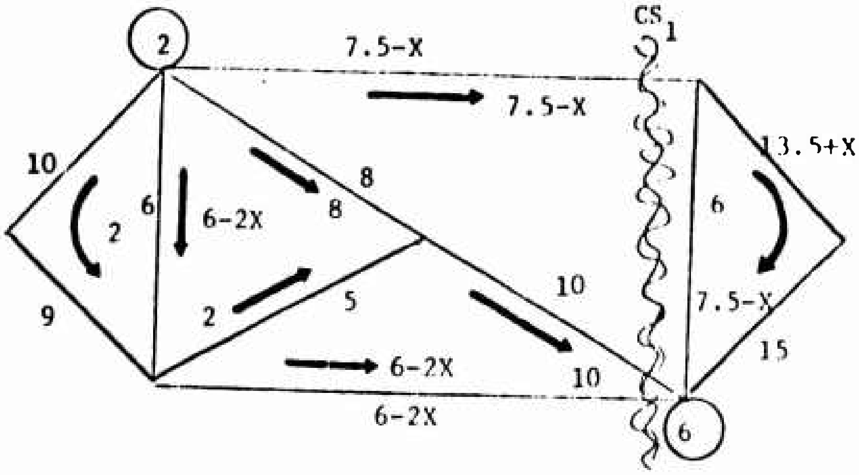
Now we get k = 3.5 for link (6,7) , it is still the smallest, so it is definitive.



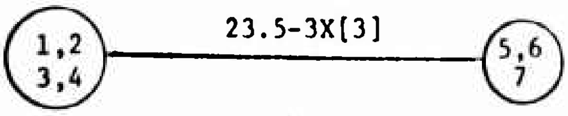
Third Cycle

$6 - 2X = 0 \longrightarrow X = 3$

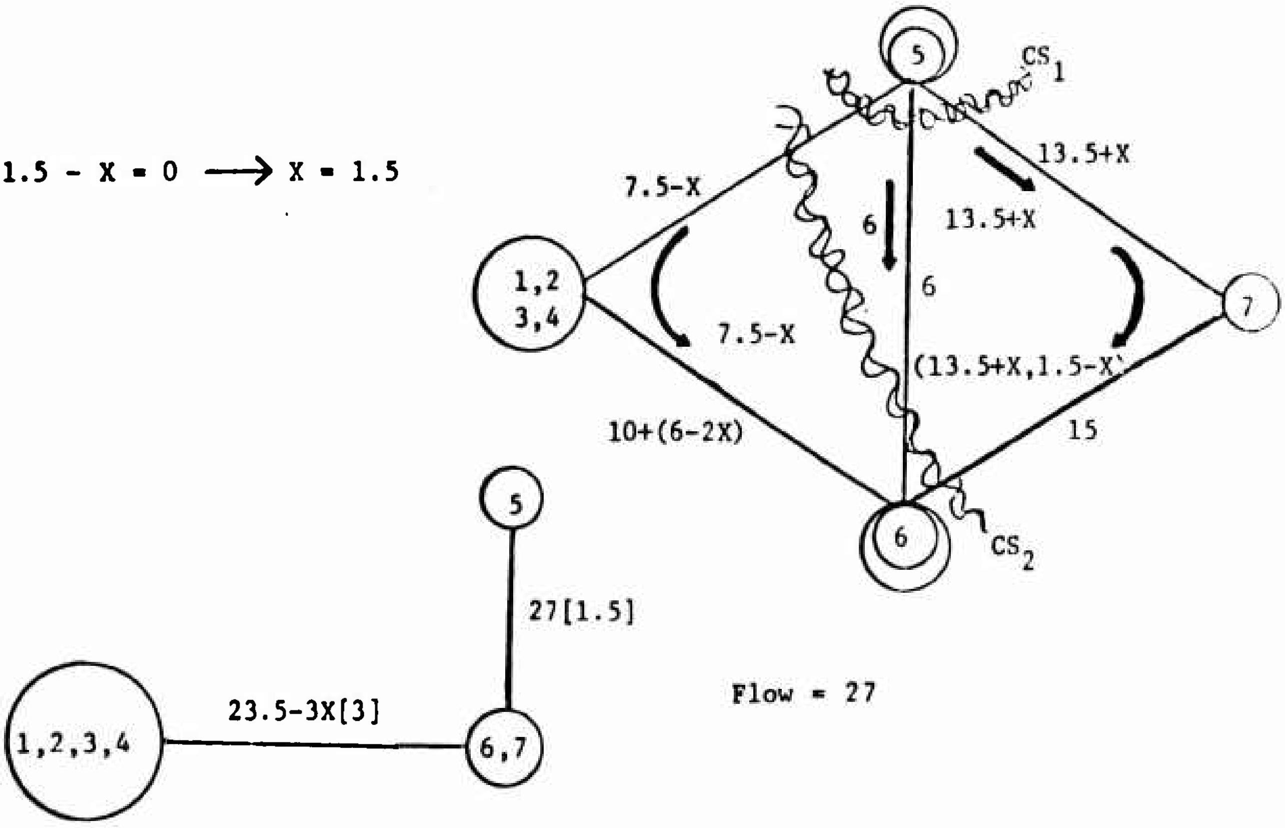
For $X > 3$ the function describing the maximum flow changes to $17.5 - X$.



Flow = 23.5 - 3X

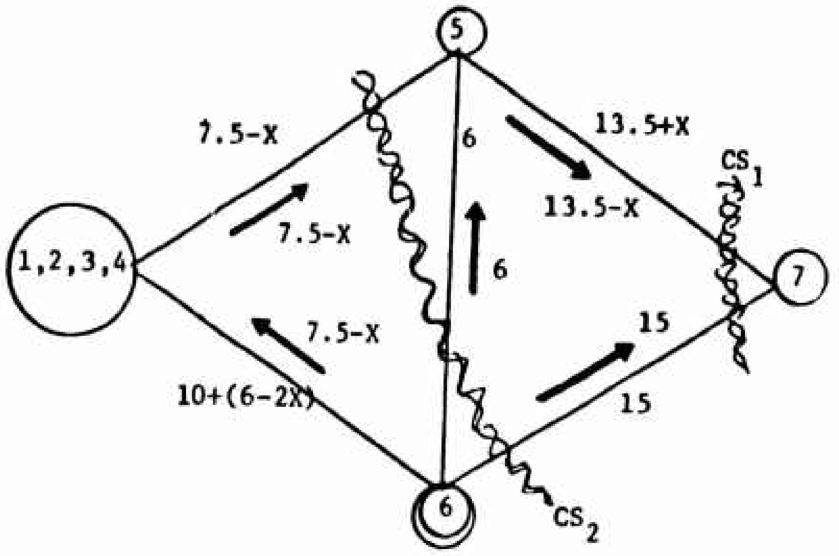


$1.5 - X = 0 \longrightarrow X = 1.5$

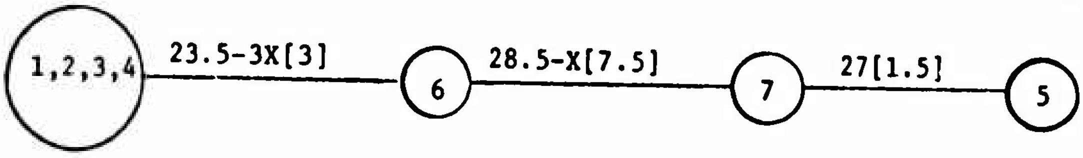


Flow = 27

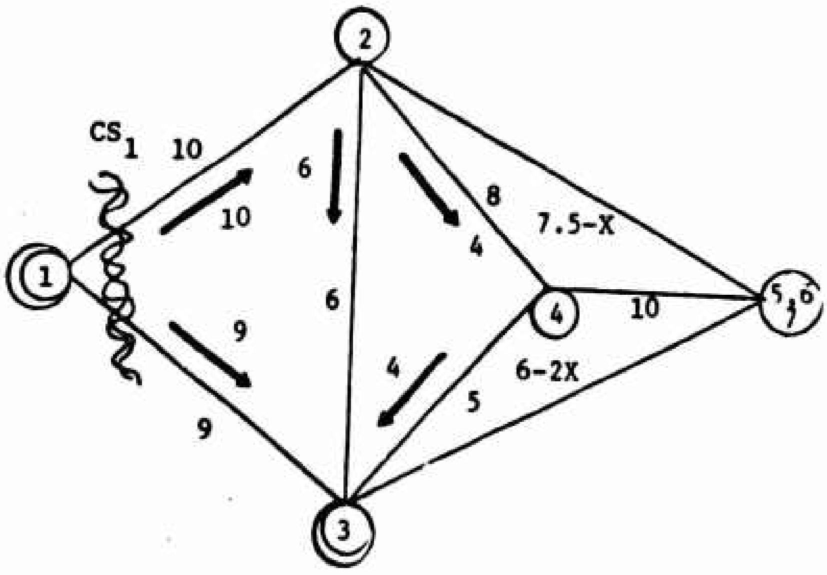
7.5-X=0 ⟶ X=7.5

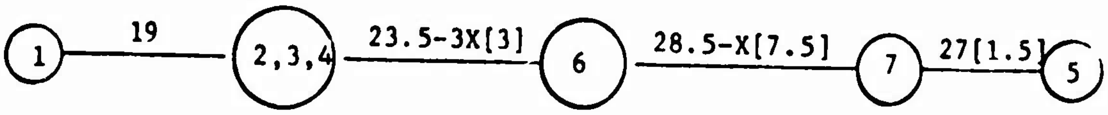


Flow = 28.5-X



No restriction

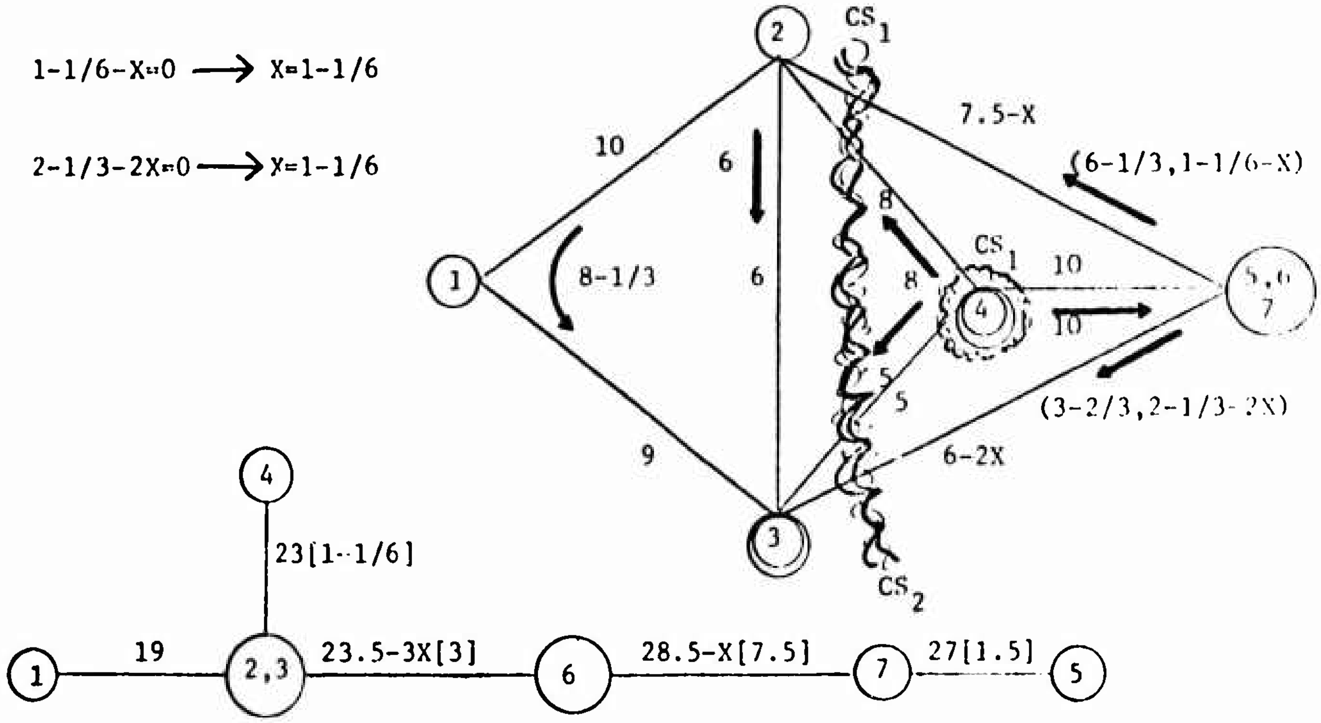




Flow = 19

1-1/6-X=0 ⟶ X=1-1/6

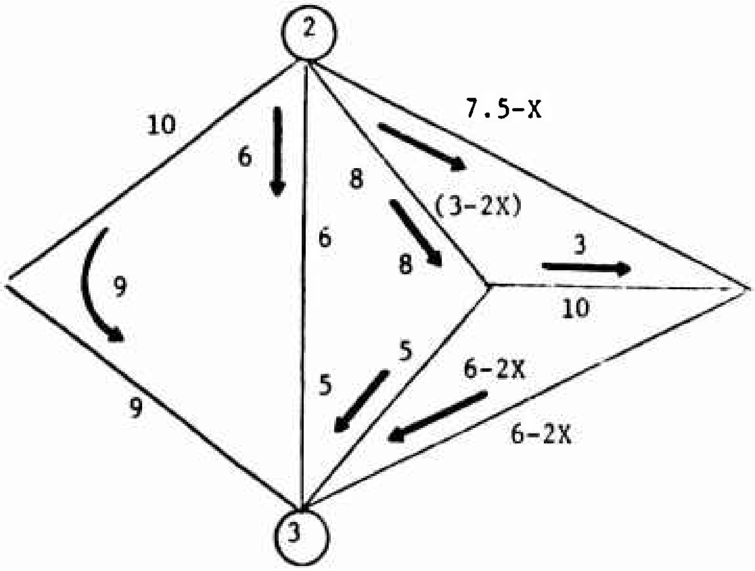
2-1/3-2X=0 ⟶ X=1-1/6

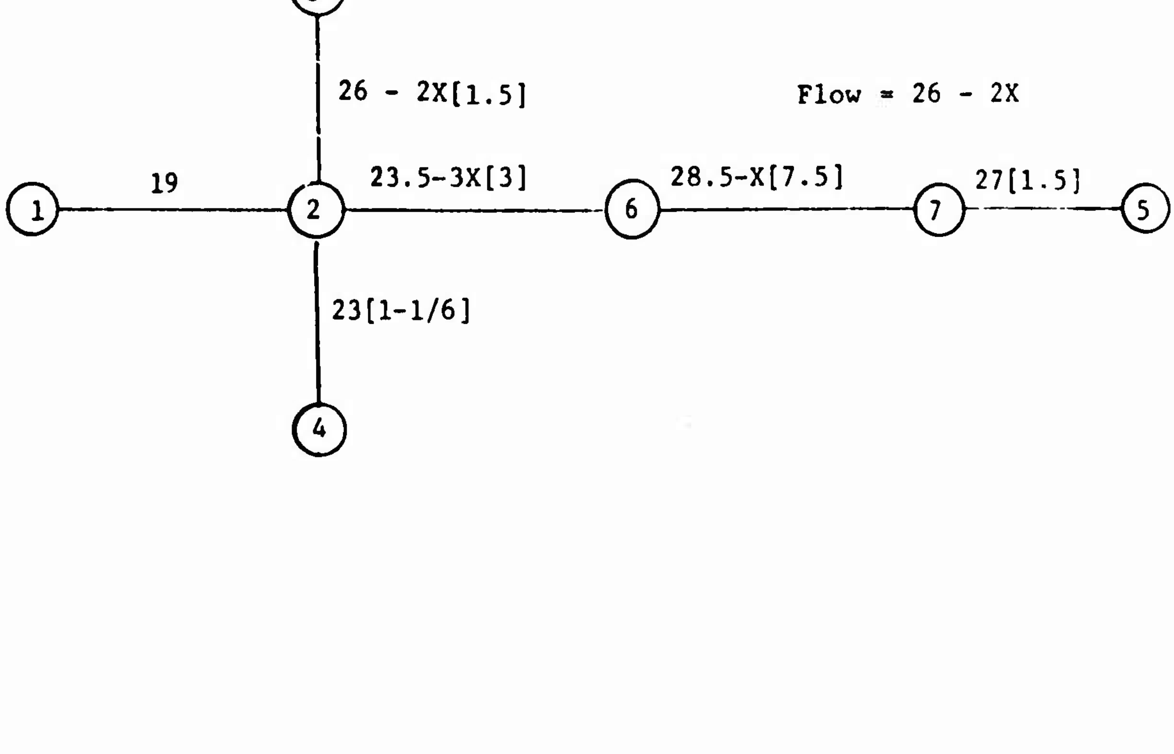


Flow = 23

3-2X=0 ⟶ X=1.5

Maximum restriction given by link (2,4) , k = 1-1/6 , it was not computed directly. It is checked in the next page.



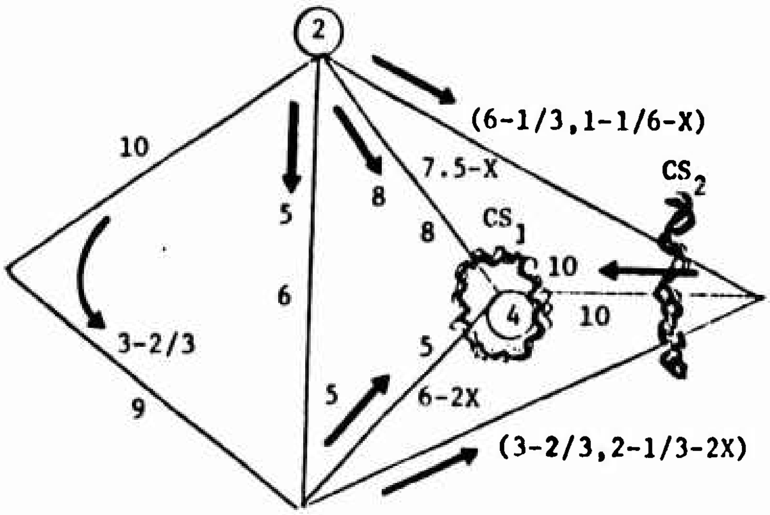


Flow = 26 - 2X

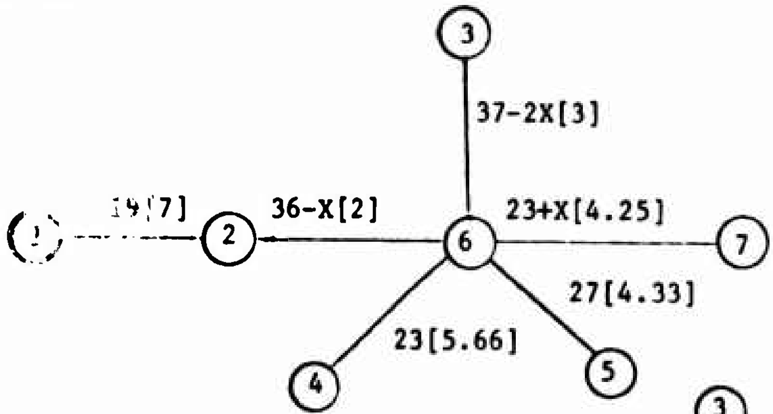
1-1/6-X=0 X=1-1/6

2-1/3-2X=0 X=1-1/6

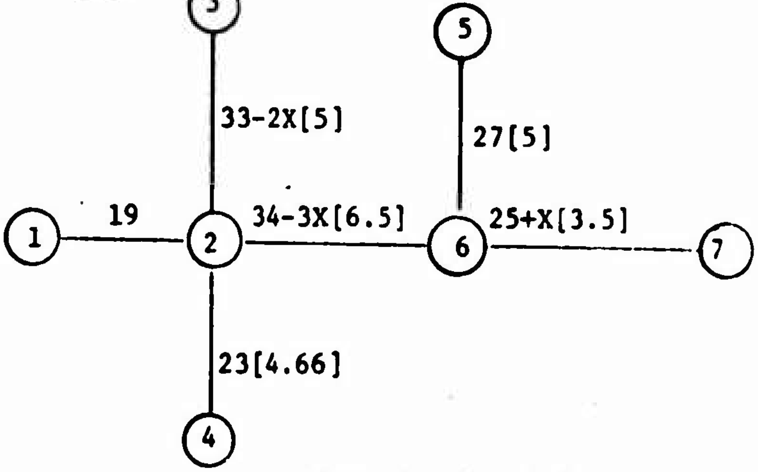
No improvement in the value of k is possible.



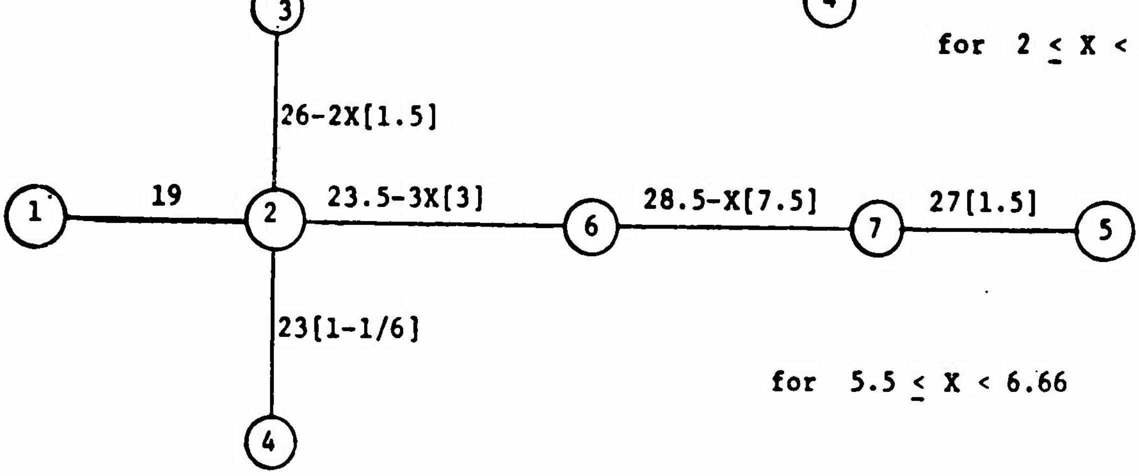
Résumé



for 0 ≤ X < 2



for 2 ≤ X < 5.5



for 5.5 ≤ X < 6.66

SECTION IV

COMMENTS, COMPARISONS

Professor Salah E. Elmaghraby of the Department of Industrial Administration of Yale University has two algorithms [2] and [4] that solve the same problems.

REFERENCES

[1] Elmaghraby, S. E., "Sensitivity Analysis of Multiterminal Flow Networks," Opns.Res., Vol. 12, No. 5, (1964).

[2] Elmaghraby, S. E., "Sensitivity Analysis of Multiterminal Flow Networks to Simultaneous Changes," Unpublished.

[3] Ford and Fulkerson, "Flow in Networks," (last chapter), Princeton University Press, (1962).

[4] Gomory, R. E. and T. C. Hu, "Multiterminal Network Flows," J.Soc.Ind. and Appl.Math., Vol. 9, No. 4, (1961).