N68-29358

No. — CR-95772

Pages - 11

Code - 0

Cat 10

ON THE CHOICE OF A 'BEST TREE' IN THE FLOWGRAPH ANALYSIS OF NETWORKS*

WILLIAM R. DUNN, JR.
Lockheed Missiles and Space Company
Sunnyvale, California

and

S. PARK CHAN
University of Santa Clara
Santa Clara, California

ABSTRACT

In the flowgraph solution of a linear electrical network, [1][2] a Mason flowgraph G_m is constructed from the linear graph G of the given network N through the selection of a tree T of G. It has been pointed out [3] that the number of loops of G_m is dependent upon the choice of such a tree T. Hence, if there exists a method for determining the "best tree" of G such that the number of loops in the corresponding Mason graph G_m is minimal, a considerable amount of computation time may be saved in the flowgraph analysis of the given N.

In this work, the problem of finding a best tree of G is discussed. First a brief discussion on the relationships between G and G_m is given, and the problem statement is formulated and simplified. Then it is shown as a special case that a star tree of G, if it exists, is a best tree.

Next, the terms "minimum distance tree" and "maximal shortest distance tree" [4] are defined and certain properties of these trees are discussed. Then, an algorithm for finding the maximal shortest distance trees of a graph is formulated and illustrative examples are given.

Finally some of the difficulties related to the problem as well as possible extension to this work are briefly discussed.

INTRODUCTION

In the flowgraph solution of linear electrical networks, [1],[2] a Mason flowgraph is constructed from the linear graph G of the network through selection of an appropriate tree of G. It has been pointed out [3] that the number of loops of G_m is dependent upon the choice of such a tree. Hence, where a tree is not properly chosen, time of computation can be significantly long.

The following paragraphs present the status of current research into the problem of finding the "best" tree of G, i.e., that tree which minimizes the number of loops of G_m.

*This work was supported in part by the National Aeronautics and Space Administration, Grant No. NGR-05-017-012.

PROBLEM STATEMENT

A linear network N comprised of passive and active elements is given with a corresponding linear graph G. The element (edge) currents $\{i_e\}$ and element (edge) voltages $\{v_e\}$ are considered to be variables of N and are represented as vertices in the Mason flowgraph G_m of N.

The edges of G_m are constructed from an element description of N and the linear graph G as follows.

First some tree T_o of G is selected such that the tree branches of T_o include all voltage driver elements in N and that the chords with respect to T_o include all current driver elements in N. (This is always possible where a unique solution exists for element currents and voltages [5]). Remaining (non-driver) branches and chords of T_o are comprised of the remaining passive elements in N. As noted, vertices of G_m correspond to element voltages and currents in N. Edges of G_m represent (1) a dummy solution edge j, (2) the V-I relation of each passive element, (3) the element interdependencies, (4) f-cutset chords with respect to T_o, and finally (5) f-circuit set tree branches with respect to T_o. Each of these edge contributions is described in detail as follows:

(1) Dummy solution edge j - Where the transfer function T(s) of network variable $e_1(s)$ over network variable $e_2(s)$ (i.e. $T(s)=e_1(s)/e_2(s)$) is the desired solution, and edge "j" is connected between the node of G_m corresponding to e_2 and the node of G_m corresponding to e_1, and directed from e_2 to e_1 .

(2) V-I relationships of passive elements - (a) Where a passive element of N corresponds to a chord of G with respect to T_o, an edge of G_m is constructed between the two nodes of G_m, one of which corresponds to the specific element voltage and the other of which corresponds to the element current of N. The edge is directed from the voltage node to the current node; (b) Passive elements of N corresponding to tree branches of G (with respect to T_o) are represented in G_m in a similar manner except that edges are directed from the current node (of G_m) to the voltage node (of G_m).

(3) Element interdependences - Where an element voltage V_j (or current I_j) of N is linearly dependent on an element voltage V_k (or current I_k), an edge is directed from the corresponding voltage (or current) node of G_m to the corresponding voltage (or current) node of G_m.

(4) f-cutset chords - Consider the f-cutsets with respect to T_o of G. The tree branch of each f-cutset corresponds to the vertex v_j of G_m which in turn corresponds to the current flowing in that branch. The chords of the f-cutset correspond to edges in G_m. For each chord, an edge of G_m is constructed such that it is directed to v_j from the node of G_m that corresponds to the current that flows through the chord. Hence an edge of G_m is obtained for each chord in each f-cutset of G.

(5) f-circuit set branches - Consider the f-circuit sets with respect to T_o of G. In following a dual argument of the preceding paragraph it is readily seen that an edge of G_m is obtained for each tree branch in each f-circuit set of G.

Given this construction procedure, the problem is to find the "best" tree of G defined as one which minimizes the number of loops in G_m. As an approach to the problem, the following simplifying assumptions are made noting that they do not seriously degrade the significance of any results obtained:

(i) The restriction that the tree contain all voltage drivers and no current drivers is disregarded.

(ii) It is assumed that G is non-oriented.

(iii) It is assumed that G has no parallel edges.

(iv) Since the number of vertices in G_m is fixed, the nullity of G_m is directly proportional to the number of edges of G_m. Intuitively, any effort to minimize nullity must <u>tend</u> to minimize the number of circuits (i.e. loops) of a graph. Hence it is assumed that the problem can in general be reduced to one of minimizing the number of edges of G_m.

MINIMIZATION OF EDGES OF G_m

In viewing the process of constructing G_m it is seen that the edges of G_m corresponding to the dummy solution j, the V-I relationships of the passive elements, and the element interdependencies are always the same in number regardless of what tree is chosen in G.

The number of edges of G_m due to the f-cutset chords and f-circuit set tree branches however varies depending upon the choice of T_o. This is now elaborated upon as follows:

In constructing G_m it was shown that each chord set of each f-cutset of G with respect to T_o corresponds to a set of edges in G_m.

Accordingly, the f cutset matrix Q_f with respect to T_o may be written thus

$$Q_f = [Q_{f_{11}} \quad U]$$

observing that the "1" entries in $Q_{f_{11}}$ correspond to chords of G with respect to T_o. Hence each "1" entry in $Q_{f_{11}}$ corresponds to an edge of G_m. The number of these edges (i.e. "1" entries in $Q_{f_{11}}$) is denoted by N_C.

In a dual manner an f-circuit matrix with respect to T_o can be written

$$B_f = [U \quad B_{f_{12}}] \; .$$

Analogously, each "1" entry in $B_{f_{12}}$ corresponds to an edge in G_m. The number of these edges of G_m is denoted by N_B.

Now using the well known relationship [5] $Q_f B_f^T = 0$ it follows at once that

$$N_B = N_C \; .$$

Hence, the problem is reduced to one of minimizing either N_b or N_c .

THE STAR TREE - A SPECIAL CASE

It is next shown that in a special case a tree T* of G exists which minimizes the number of edges in G_m. First, the following terms are defined.

Definition 1 - A star tree T* of a finite, connected, linear graph G (containing no parallel edges) is a tree of G which has a vertex that is connected to each of the other vertices of the graph by exactly one branch.

Definition 2 - The central vertex of a graph G having star tree T* is that vertex of T* which is connected to every other vertex of T*.

Note that the star tree is not necessarily unique to a graph as can be seen by considering the case of the complete graph.

Next, the quantities N_b and N_c which were introduced in the previous section are more rigorously defined.

Definition 3 - Given a tree T_o of the graph G. The total number of chords that appear in all the cutsets of G with respect to T_o is defined as the chord count of T_o and its magnitude denoted by positive integer N_c. It should be pointed out that N_c will always include repeated chords.

Definition 4 - Given a tree T_o of the graph G, the total number of tree branches that appear in all the f-circuit sets of G with respect to T_o

is defined as the branch count of T_o, and its magnitude is denoted by positive integer N_b. As in the case of N_c, N_b will include repeated edges.

The following lemma may now be stated.

Lemma 1 - Given the linear, connected graph G of n vertices and e edges containing star tree T^* with central vertex V_c. The number of edges of the corresponding flowgraph G_m of G is a minimum when G_m is formed using T^*.

Proof: The lemma is proved by contradiction. First, assume that the star tree T^* of G exists as shown in Figure 1, where V_c is the central vertex of T^*.

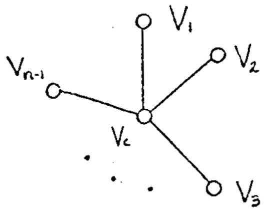

Figure 1

Since G has no parallel edges, the chords of G with respect to T^* connect only vertices $\{V_1, V_2, V_3, \ldots, V_{n-1}\}$. Note that each chord forms an f-circuit having exactly two tree branches such that the branch count N_b is exactly $2(e-n+1)$.

Now assume that a tree T' exists having a branch count which is smaller than $2(e-n+1)$. Then there will be at least one f-circuit of G with respect to T' which contains only one tree branch. But the f-circuit contains exactly one chord which means that the tree branch and the chord are parallel edges which is impossible since there are no parallel edges in G. Therefore, no tree T' exists meaning that T^* has the minimum branch count. Hence the lemma.

The utility of this lemma is limited since a graph may not necessarily have a star tree. Consequently, the problem is to determine the "best" tree for the general case.

THE GENERAL CASE

In finding the "best" tree for the general case the problem can be viewed in terms of choosing a tree of G that minimizes the branch count N_b or the chord count N_c. Since $N_b = N_c$, we confine our attention to minimizing N_b. Now the problem at hand can be dealt with by considering the properties of the kth f-circuit matrix $B_f^{(k)}$ in terms of the graph G. In doing so, the distance between vertices of a non-oriented graph is first defined.

Definition 5 - The distance $d(V_j,V_k)$ between two vertices V_j and V_k of a non-oriented, connected, non-seperable graph G is the number of edges in the shortest path connecting vertices V_j and V_k. Writing the kth f-circuit matrix,

$$B_f^{(k)} = [U \quad B_{f_{12}}^{(k)}]$$

it is known [5][6] that each column of U corresponds to an edge of G that is a chord with respect to some tree T_o of G whereas each column of $B_{f_{12}}^{(k)}$ corresponds to an edge of G that is a tree branch of T_o. Now examine a given row of $B_f^{(k)}$. The "1" entry of the row that is in sub-matrix U corresponds to a chord connecting vertices V_j and V_k. The "1" entries in $B_{f_{12}}^{(k)}$ in the same row correspond to tree branches in the (unique) tree path connecting V_j and V_k. Now since there is only one such path in T_o (otherwise we would have a circuit and T_o would not be a tree) the number of "1" entries in the row of $B_{f_{12}}^{(k)}$ is clearly the distance $d(V_j,V_k)$ between vertices V_j and V_k in T_o. Now if the distance between each vertex pair corresponding to each chord of G with respect to T_o is calculated and these distances for all chords are summed, the total number is certainly the branch count N_b (i.e. the number of "1" entries in $B_{f_{12}}^{(k)}$). The problem can now be stated as one in which a minimum distance tree, defined as follows, is sought.

Definition 6 - Let V_j and V_k be the vertex pair connecting a chord of G with respect to T_o and let $d(V_j,V_k)$ be the distance between vertices V_j and V_k in T_o. T_o is a minimum distance tree where the quantity

$$\sum_{\left\{\substack{\text{all chords of G} \\ \text{with respect to } T_o}\right\}} d(V_j,V_k)$$

is a minimum.

Now finding the minimum distance tree appears related to the finding of the maximal shortest distance tree which has been solved in connection with the "shortest route" problem [4].

THE SHORTEST ROUTE PROBLEM

Basically, the "shortest route" problem is that of finding the shortest directed path between two specified vertices in a weighted (i.e. edges have weights assigned to them) digraph. In terms of the non-oriented, unweighted (i.e. each edge has "weight" of unity), connected graph G in the problem at hand the shortest route problem is simply that of finding the distance between the specified vertices V_j and V_k. It has been pointed out [4] that in the process of finding the distance between V_j and V_k the distance between V_j and every other vertex of G is obtained. Consequently, in solving the shortest route problem one solves the more general problem of finding the distance between some reference vertex V_o and every other vertex in G. (An original algorithm for doing this is given at the end of this paper.)

Definition 7 [4] - A maximal shortest distance tree relative to vertex V_o of G is a tree such that a path from V_o to each of the other vertices of G contains a minimum number of edges.

Example #1 - Consider the graph G of Figure 2a with reference vertex V_o. Solid edges of Figures 2b and 2c are maximal shortest distance trees with respect to vertex V_o.

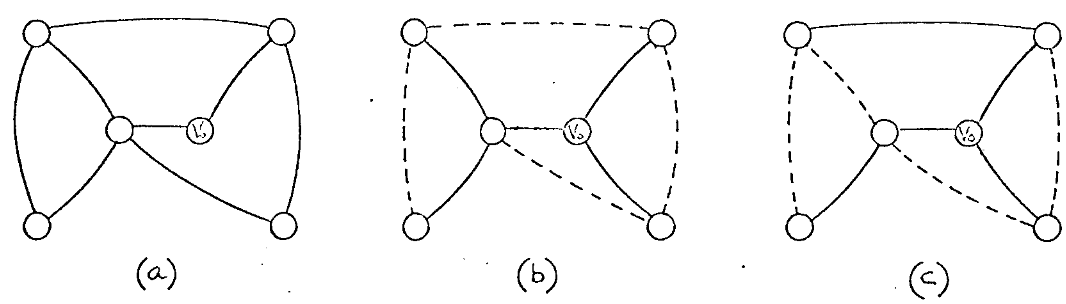

Figure 2

This example reveals two important properties of the maximal shortest distance tree: (a) A maximal shortest distance tree with respect to a vertex V_o is not necessarily unique; (b) The maximal shortest distance trees with respect to V_o (if there is more than one) need not have the same branch count. For example, in Fig. 2b, $N_b = 9$; in Fig. 2c, $N_b = 11$.

These properties are important to the problem of this paper to the extent that any method that yields a maximal shortest distance tree may not necessarily yield a minimum distance tree.

On the other hand, a minimum distance tree need not be a maximal shortest distance tree as can be seen in the graph of Fig. 3.

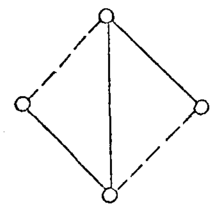

Figure 3

That there is at least one maximal shortest distance tree that is a minimum distance tree has yet to be proven or counterexampled in the research conducted to date.

The association of these two trees on the other hand clearly exists for the case of the star tree as summarized in the following lemma.

Lemma 2 - A star tree is a maximal shortest distance tree.

On the strength of this association, the following algorithm for the generation of all maximal shortest distance trees of a graph is presented with the qualification that although the method has yielded a minimum distance tree in the examples tested to date, no formal proof has been made of its validity.

AN ALGORITHM FOR FINDING THE MAXIMAL SHORTEST DISTANCE TREES OF A GRAPH

To obtain the maximal shortest distance trees of a graph we make use of the well known connection matrix [6]. The connection matrix is related to the problem of this paper in the following definition and theorem.

Definition 8 - Given a non-oriented graph G of V vertices, a connection matrix $C = [c_{ij}]$ has rows and columns corresponding to the vertices of G where the entry c_{ij} symbollically denotes the edge e_{ij} connected between vertices V_i (row of C) and V_j(column of C). Graph G is assumed to have no self loops (i.e. $e_{ii} = 0$). Likewise, since G is non-oriented, $e_{ij}=e_{ji}$.

The following well known theorem [6] is restated for clarity of discussion.

Theorem 1 - The (i,j) entry of $C^r = C \cdot C \cdots C$ (r factors) is in general a summation of products each of which symbollically corresponds to the product of r edges $e_{i\alpha}$, $e_{\beta\gamma}$, ... $e_{\gamma j}$ that form a path between vertex i and j. Note that a path may constitute the retracing of one or more edges.

Example #2 - Consider the graph of Figure 4.

$$C = \begin{array}{c c} & \begin{array}{ccc} V_1 & V_2 & V_3 \end{array} \\ \begin{array}{c} V_1 \\ V_2 \\ V_3 \end{array} & \begin{bmatrix} 0 & e_{12} & e_{13} \\ e_{12} & 0 & e_{23} \\ e_{13} & e_{23} & 0 \end{bmatrix} \end{array}$$

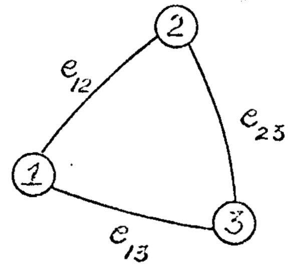

Figure 4

$$C^2 = \begin{bmatrix} (e_{12}e_{12}+e_{13}e_{13}) & e_{13}e_{23} & e_{12}e_{23} \\ e_{13}e_{23} & (e_{12}e_{12}+e_{23}e_{23}) & e_{12}e_{13} \\ e_{12}e_{23} & e_{12}e_{13} & (e_{13}e_{13}+e_{23}e_{23}) \end{bmatrix}$$

Now clearly, the path length represented by the nonzero $(ij)^{th}$ element of C^k is shorter than that of C^{k+1} . For this reason, the following is stated as an algorithm for listing the maximal shortest distance trees of a graph.

Lemma 3 - Given a linear, connected non-oriented graph G with connection matrix C. Element entries in C are each represented symbollically by e_{ij} which denotes the edge joining vertices i and j. The maximal shortest distance trees with respect to vertex i (rows of C) is determined through application of the following steps to all the rows of C:

Step 1. Examine the entries in row i of C. Note the set of non-zero entries and the set of solumns S_1 in which they occur. Retain these edges as branches of a maximal shortest distance tree calling the set K_1.

Step 2. Form C^2 and examine entries in row i of C^2, ignoring entries corresponding to the set of columns S_1. Note the set of non-zero entries and the set of columns S_2 in which they occur. Remove edges in S_2 that appear in K_1 and call the resulting set of edges K_2. Find all possible sets $K_{21}, K_{22}, \ldots, K_{2\ell}$ that can be formed by choosing one edge of K_2 from each column of S_2 for all columns of S_2. Note that edges appearing in element c_{ij} of C^2 where i = j are contained in K_1. Hence $K_{2\ell}$ will always contain a fewer number of edges than there are columns in S_2.

Step 3. In general, for any positive integer (k-1), given C^{k-1} form C^k and examine entries in row i of C^k ignoring entries corresponding to $S_1, S_2, \ldots, S_{k-1}$.

Note the set of non-zero entries and columns S_k in which they occur.

Remove edges in columns S_k that appear in $K_1, K_2, \ldots, K_{k-1}$. Call the resulting set K_k . Find all (ℓ) possible sets $K_{k1}, K_{k2}, \ldots, K_{k\ell}$ which can be formed by choosing one edge from each column S_k . Combine each of these with each previously generated set. Call the resultant set

$$\{K_1,K_{21},K_{31},K_{k1}\}, \ldots, \{K_1,K_{21},\ldots,K_{k2}\}, \ldots, \{K_1,K_{21},\ldots,K_{j2},\ldots,K_{k1}\},$$
$$\ldots,\{K_1,K_{21},\ldots,K_{j\ell},\ldots, K_{k1}\}, \ldots, \{K_1,K_{2\ell},K_{3\ell},\ldots,K_{k\ell}\}.$$

Step 4. Continue this process until (v-1) vertices have been eliminated. The resultant set produced provides the maximal shortest distance trees with respect to vertex i.

Step 5. Perform steps 1 through 4 for all rows of C. The resultant, non-redundant sets of edges generated are all the maximal shortest distance trees of the graph.

Example #3 - Given the graph of Figure 5. Consider finding the maximal shortest distance tree with respect to V_6.

$$C = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{array} \begin{array}{c} \begin{array}{cccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \end{array} \\ \begin{bmatrix} 0 & e_{12} & e_{13} & e_{14} & 0 & 0 \\ e_{12} & 0 & e_{23} & e_{24} & 0 & 0 \\ e_{13} & e_{23} & 0 & e_{34} & 0 & e_{36} \\ e_{14} & e_{24} & e_{34} & 0 & e_{45} & e_{46} \\ 0 & 0 & 0 & e_{45} & 0 & e_{56} \\ 0 & 0 & e_{36} & e_{46} & e_{56} & 0 \end{bmatrix} \end{array}$$

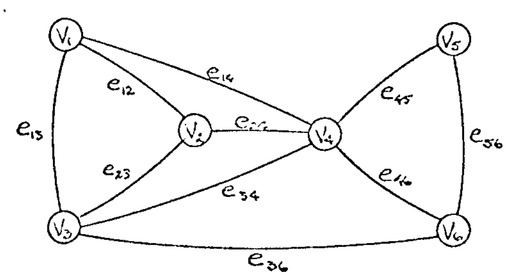

Figure 5

Step 1. $S_1 = \{V_3, V_4, V_5\}$; $K_1 = \{e_{36}, e_{46}, e_{56}\}$

Step 2. The last row (V_6) of matrix C^2 is

$$V_6[\overset{V_1}{(e_{13}e_{36}+e_{14}e_{46})}\ \overset{V_2}{(e_{23}e_{36}+e_{24}e_{46})}\overset{V_3}{(e_{34}e_{46})}\overset{V_4}{(e_{34}e_{36}+e_{45}e_{56})}$$
$$\overset{V_5}{(e_{45}e_{46})}\ \overset{V_6}{(e_{36}^2 + e_{46}^2 + e_{56}^2)}] .$$

$S_2 = \{V_1,V_2,V_6\}$, $K_2 = \{e_{13},e_{14},e_{23},e_{24}\}$;

$K_{21} = \{e_{13},e_{23}\}$, $K_{22} = \{e_{13},e_{24}\}$, $K_{23} = \{e_{14},e_{23}\}$, $K_{24} = \{e_{14},e_{24}\}$.

(Note that K_{21}, K_{22}, K_{23}, and K_{24} have only 2 edges whereas S_2 has 3 columns. This is because the edges appearing in the column corresponding to V_6 are contained in K_1.)

Resulting set is

$$\{K_1, K_{21}\} = \{e_{13}, e_{23}, e_{36}, e_{46}, e_{56}\} \quad \{K_1, K_{22}\} = \{e_{13}, e_{24}, e_{36}, e_{46}, e_{56}\}$$

$$\{K_1, K_{23}\} = \{e_{14}, e_{23}, e_{36}, e_{46}, e_{56}\} \quad \{K_1, K_{24}\} = \{e_{14}, e_{24}, e_{36}, e_{46}, e_{56}\}$$

Process terminates since $v-1=5$ vertices have been eliminated. Hence resultant set is the set of all maximal shortest distance trees with respect to v_6.

Example #4 - Consider finding the maximal shortest distance tree with respect to V_4 of the graph of Figure 5.

Step 1. C is as given in Example 3.

$$S_1 = \{V_1, V_2, V_3, V_5, V_6\}, \quad K_1 = \{e_{14}, e_{24}, e_{34}, e_{45}, e_{46}\}$$

Process terminates since $v-1=5$ vertices have been eliminated.

Maximal shortest distance tree with respect to V_4 is clearly a star tree.

CONCLUSION

In the foregoing, the topological relationships between the linear graph G of network N and the corresponding Mason flowgraph G_m are established. Under the noted simplifying assumptions it is shown that the star tree, where it exists, is the "best" tree, i.e., that tree which minimizes the complexity of G_m.

It is established that the "best" tree is a minimum distance tree which for the case in which a star tree exists is one of a set of maximal shortest distance trees. An algorithm for deriving the latter tree is presented.

The association between the minimum distance tree and maximal shortest distance tree is however not established for the general case and accordingly remains as an unsolved problem.

REFERENCES

[1] F.M. Verduyn and S.P. Chan, "Computer Evaluation of Network Performance by a Topological Flowgraph Technique, Asilomar Conference, Nov. 1967.

[2] W.W. Happ, "Flowgraph Techniques for Closed Systems," IEEE AES Trans., AES-2, No. 3, pp 252-264, May 1966.

[3] W.W. Happ, "Flowgraphs as a Teaching Aid," IEEE Trans. on Education, Vol. E-9, No. 2, pp 69-80, June 1966.

[4] R. Busacker and T. Saaty, Finite Graphs and Networks: An Introducation with Applications, McGraw-Hill, 1965.

[5] S.P. Chan, Introductory Topological Analysis of Electrical Networks to be published by Holt, Rinehart and Winston, 1966.

[6] Seshu and Reed, Linear Graphs and Electrical Networks, Addison-Wesley, 1961.