On The Determination of a Low Cost K-Connected Network

S.Latha, Dr. S. Senthil kumar
Associate Professor &Head, Dept. of Electronics & Instrumentation Engineering, Bharath University, Chennai, Tamil Nadu, India
Associate Professor, Dept. of Electronics & Instrumentation Engineering, Bharath University, Chennai, Tamil Nadu, India

ABSTRACT: The goal of the topological design of a computer network is to achieve a specified performance at a minimal cost. Unfortunately, the problem is completely intractable. A reasonable approach is to generate a potential network topology. One heuristic for generating a potential network topology is due to Steiglitz, Weiner and Kleitman and is called the link deficit algorithm. This paper presents a modified version of the above heuristic which results in a lower cost starting network. In addition, a novel and simple algorithm has been presented for designing a 3-connected starting network when the number of nodes in the network exceeds four.


1. INTRODUCTION

The terms ‘K-connected network’, ‘starting network’, ‘link deficit algorithm’, ‘topological design’, ‘minimal spanning tree’ have the usual meanings[1,2,3].

The geographical positions of the nodes of a network are given. A cost matrix gives The cost of establishing a link between any node i to any node j (i is not equal to j). The minimal spanning tree is determined using Kruskal’s algorithm or Prim’s algorithm [2].Now the link deficit algorithm is applied to obtain the starting network. We illustrate this novel algorithm by working out an illustrative example.[7,8]
III. ILLUSTRATIVE EXAMPLE

The geographical positions of five nodes labeled 1 through 5 are shown in fig 1.

![Fig 1](image)

The cost of establishing a link between any nodes i to any node j is given by the following cost matrix $C$

The minimal spanning tree can be determined as usual [2] and is shown in fig 2. Let us suppose that we want to set up a 3-connected network. In fig 2, the deficits of the nodes 1 through 5 are 0, 2, 2, 1 and 2 respectively. Application of the link deficit algorithm to what is shown in fig 2 results in the starting network shown in fig 3.

$$C = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 \\
1 & 2 & 2 & 1 & 8 \\
2 & 2 & 4 & 10 & 5 \\
3 & 2 & 4 & - & 6 \\
4 & 1 & 10 & 4 & - \\
5 & 8 & 5 & 6 & 2 \\
\end{pmatrix}$$

(1)

This modified link deficit algorithm can be used to set up any k-connected network wherein $K$ is greater than 1. We now present a simple algorithm. Wherein the minimal spanning tree algorithm is applied twice.[11]
IV. ALGORITHM FOR A 3-CONNECTED NETWORK

Given the geographical positions of the nodes of a network and a cost matrix giving the cost of establishing a link between any node i and any node j, we proceed as follows. The minimal spanning tree T1 is determined. The costs of the link presented in T1 are made infinite in the cost matrix and the minimal spanning tree T2 is again determined. A low cost 3-connected starting network results when the edges of T1 and T2 are superimposed. We now illustrate this novel approach by working out the previous example.

V. ILLUSTRATIVE EXAMPLE

The geographical positions of the nodes of a network are shown in fig 1. The cost matrix is given by equation 1. The minimal spanning tree is determined as usual and is shown in fig 2. In the cost matrix given in equation 1, the costs of the edges present in fig 1 are made infinite, and the modified cost matrix C’ is shown below.

\[
C' = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 \\
\infty & \infty & \infty & 8 & \\
\infty & \infty & \infty & 5 & 10 \\
4 & \infty & \infty & 6 & 4 \\
5 & 8 & 6 & \infty & \infty \\
\end{pmatrix}
\]

For the cost matrix C’ given in equation 2, we find the minimal spanning tree as shown in fig 4. When the edges shown in fig 2 and fig 4 are superimposed, we get a low cost 3 connected starting network as shown in fig 5.[9][10]
VI. CONCLUSION

This paper presents a modified version of the link deficit algorithm which results in a lower cost starting network, when we desire to set up a k-connected network. In particular when k=3, a novel algorithm which utilizes the minimal spanning tree algorithm twice has been presented.

REFERENCES