Network Structures for Economic Communication

L.T.M. Berry

Centre for Teletraffic Research, Bond University, Gold Coast, Queensland, Australia

The problem of finding an optimal tree structure to carry traffic between nodes using minimal channel capacity has previously been formulated by Hu as a constrained graph problem. A new practical solution is described which is suitable for large networks.

The basis of the algorithm is a decomposition result which solves the problem of identifying nodes in disjoint tree structures which, when joined satisfy the above minimization criterion.

The decomposition and optimal interconnection method proposed may be applied to other generalizations of the network synthesis problem.

1 Introduction

Trees are connected graphs without cycles. In this paper we will refer to the components of a graph $G[N, L]$ as nodes and links where $N = \{1, 2, \ldots, i, \ldots, n\}$ and $L \subseteq \{(i,j) | i,j \in N\}$. When a link exists between every pair of nodes the graph is said to be complete. We will be interested in associating with each node pair $(i,j)$ a number $t(i,j)$, the traffic value from node $i$ to node $j$. In general, $t(i,j) \neq t(j,i)$ and $(i,j)$ need not be a link of the graph. In an undirected connected graph, a sequence of links $(i_1, i_2), (i_2, i_3), \ldots, (i_{n-1}, i_n)$ is called a path, provided the nodes in the sequence are distinct. Clearly, the number of links in an $n$-node tree is $n - 1$. As indicated, we will be concerned with labelled trees.

Tree structures play an important role in communications. For example, questions of the following kind are likely to involve trees:

- given a set of nodes and their traffic demands $t(i,j)$, what is the best way to interconnect them? The nodes themselves may represent subnetworks.
- a group of nodes in a network wish to belong to a multicast, how should the traffic routes be selected?
- given a network $G$ find a minimum traffic weighted spanning tree containing selected nodes in $G$, other nodes may also be included.

In an earlier paper Gomory and Hu [2] gave algorithms for constructing a cut-tree in polynomial time. Thus the problem is not NP-complete. It is worth pointing out that although the $t(i,j)$ are directed, the problem is essentially one of an undirected network (see section 4 equation (9) for details).

The motivation for considering this (solved, in principle) problem was two-fold. Firstly, the implementation of Hu’s algorithm requires the solution of $(n - 1)$ maximal flow problems, each taking at most $O(n^3)$ applications of the Ford-Fulkerson algorithm [3] (or an equivalent algorithm). Thus the algorithm is $O(n^4)$. Because of some technical problems Dinic’s algorithm [4] was used instead of the Ford-Fulkerson algorithm in our implementation of this approach. Some inconsistent results were obtained, the ‘optimal’ value generated being dependent on the declared origin and destination nodes used to start the algorithm. It seemed worthwhile to attempt to find a reliable fast approach, suitable for practical application to large networks. Secondly, it was thought that methods developed for this problem may generalize to other constrained network problems of interest which are NP-complete.

2 Description of the Optimum Communication Spanning Tree Problem

Cayley [5] proved that, for an $n$-node complete graph, the total number of spanning trees is $n^{n-2}$ ($n \geq 2$). Therefore, starting with $n$ nodes there are exactly $n^{n-2}$ distinct trees that could be constructed to carry traffic between the node pairs. For each of these we may compute the sum of the link flows carried on the network. Our aim is to identify the subset of trees which minimize the sum of the link flows. This quantity is directly related to the total network channel
capacity required to carry the traffic. We now formulate the problem.

Let \( d(i, j) \) denote the distance between nodes \( i \) and \( j \). That is, the minimum number of links between nodes \( i \) and \( j \). Given the traffic demands \( t(i, j) \) between nodes \( i \) and \( j \)

\[
Z^* \equiv Z(T^*) = \min_{T \in \mathcal{T}} \sum_{i,j} t(i, j)d(i, j)
\]  

That is, from the set of \( n^n - 2 \) trees \( T \) find \( T^* \) which minimizes the sum of the link flows. In (1) we are simply forming a weighted sum of the traffics with weights that are the number of links separating node pairs.

It is useful to give an alternative formulation. For each node \( i \) and a given tree \( T \), we determine its contribution, \( Z_i(T) \) to the link flows on the tree where

\[
Z_i(T) = \sum_j t(i, j)d(i, j)
\]

For a given tree \( T \), the total link flow \( Z(T) \) is calculated from

\[
Z(T) = \sum_{i=1}^n Z_i
\]

Our problem is to solve

\[
Z(T^*) = \min_{T \in \mathcal{T}} Z(T)
\]

We note that the solution to (4) may not be unique. Also, exhaustive enumeration is not a possibility for \( n \) much greater than about 10 (\( 10^8 \) trees to generate and compare), even though efficient algorithms exist for systematically generating all spanning trees of a complete graph [6].

Optimal communication spanning trees carry the traffic with minimal total link channel capacity requirements. Many variations on the main problem can be formulated. For example, restriction on maximum link capacity, inclusion of link costs or reliability considerations could be introduced.

In the following sections we will begin with a simple heuristic and then examine ways to improve on our solution.

3 A Greedy Algorithm Heuristic

The example with traffic matrix \( A \) (below) is used to illustrate the approach. In order to obtain the total traffic demands between origin-destination pairs we form the matrix \( F = A + A^T \).

\[
A = \begin{bmatrix} 0 & 10 & 4 & 5 \\ 13 & 0 & 6 & 7 \\ 1 & 5 & 0 & 4 \\ 14 & 8 & 7 & 0 \end{bmatrix}
\]

\[
F = \begin{bmatrix} 0 & 23 & 5 & 19 \\ 23 & 0 & 11 & 15 \\ 5 & 11 & 0 & 11 \\ 19 & 15 & 11 & 0 \end{bmatrix}
\]

Heuristic 1:

Using the elements \( f_{ij} \) as weights on the links \((i, j)\) of a complete graph, \( G \), we find the maximum spanning tree of \( G \).

For this example, we start by connecting nodes 1 and 2 (max. weight of 23). We next connect node 4 to node 1 (weight 19) then finally either node 3 to node 2 or node 3 to node 4 (cycles are avoided when forming the spanning tree). This process results in the trees illustrated in Fig 2.

![Greedy heuristic solutions](Z = 126)

Fig 2 Greedy heuristic solutions (Z = 126)

The (non-optimal) solutions found by the greedy heuristic have a \( Z \) value less than 6% greater than the minimum value \( Z^* = 119 \). The motivation for the method is simply to place nodes with largest traffic interaction close to each other on the tree. For this example, the worst solution has \( Z = 160 \) (an increase of approximately 34% above that of the optimal value).

We note that the heuristic is fast and applicable to very large networks. The principal issues to resolve are: 1) how close is it to the optimal solution? and 2) can one improve on this method.

In the following section we examine the problem of finding a lower bound for the optimal value, \( Z^* \).

4 Lower Bounding \( Z^* \)

An obvious lower bound on \( Z^* \) is obtained from (1) by noting that \( d(i, j) \geq 1 \) \( \forall i, j \).

Thus

\[
Z^* \geq \sum_{i=1}^{n} \sum_{j=1}^{n} t(i, j)
\]

For the traffic matrix \( A \) this gives \( Z^* \geq 84 \), whereas the value of \( Z^* \) is 119. We now proceed to find a ‘tighter’ lower bound. It is convenient to introduce the upper triangular matrix \( U \), consisting of the elements \( f_{ij}, j > i \) and zero elsewhere. For the example discussed above
We observe that the optimal value \( Z^* \) (and the optimal structure) only depends on the sums \( u_{ij}(=t(ij)+t(ji)), j > i \) and not the individual values \( t(ij), t(ji) \). Thus, we can assume uni-directional flows of magnitude \( U_{ij} \) (since \( d(i,j) = d(j,i) \)).

From (1)

\[
Z^* = \min_{T \in \mathcal{O}} \sum_{i=1}^{n} \sum_{j=1}^{n} t(i,j)d(i,j) \\
= \min_{T \in \mathcal{O}} \sum_{i=1}^{n} \sum_{j=i+1}^{n} u_{ij}d(i,j) 
\]

That is,

\[
Z^* = \min_{T \in \mathcal{O}} \sum_{i} \sum_{j > i} u_{ij}d(i,j) 
\]

(9)

Thus, we can assume uni-directional flows of magnitude \( u_{ij}(j > i) \) from node \( i \) to node \( j \), replacing the bi-directional traffic flows. We shall refer to the \( u_{ij} \) as modified O-D traffic values. Next, we observe that for any tree \( T \) there are exactly \( n(n-1)/2 \) modified O-D traffic values and exactly \( (n-1) \) values of \( d(i,j) \) have the value 1; other distances are \( \geq 2 \).

Lower Bound for Arbitrary Trees

Ordering the \( u_{ij} \) in non-increasing order: \( u_1, u_2, \ldots, u_{n(n-1)/2} \) it is clear that

\[
Z^* \geq \sum_{i=1}^{n-1} u_i + \frac{n(n-1)/2}{i=n} u_i 
\]

(10)

For the traffic matrix (5) we obtain the improved lower bound \([u = (23, 19, 15, 11, 11, 5)]\)

\[
Z^* \geq 1(23 + 19 + 15) + 2(11 + 11 + 5) \Rightarrow Z^* \geq 111
\]

Recall that the previous lower bound was 84 and \( Z^* = 119 \).

5 The Optimal Star Network

The computational time required to apply heuristics similar to the greedy heuristic is negligible compared with exhaustive enumeration of the \( n^{n-2} \) trees (for \( n \) large). Therefore, we can afford to provide a number of heuristics and to accept the best solution found. One class of spanning trees for which analytical results can be obtained are the star networks. These consist of a central 'hub' node to which the remaining \( n - 1 \) nodes are directly attached.

Proposition 1: If the offered traffics are equal the solution to (1) is a star network with any one of the \( n \) nodes selected as the hub node. Without loss of generality, let \( t(i,j) = 1 \forall i, j(i \neq j) \).

Proof:

\[
Z^* = \min_{T \in \mathcal{O}} \sum_{i} \sum_{j > i} u_{ij}d(i,j) \\
= \min_{T \in \mathcal{O}} \sum_{i} \sum_{j=1}^{n} d(i,j) \\
= \sum_{i=1}^{n-1} 2 + \sum_{i=n}^{n(n-1)/2} 2 
\]

(for a star network)

\[
= 2(n-1)^2 
\]

which is equal to the lower bound (10) \( t(i,j) = 1 \forall i, j \Rightarrow u_{ij} = 2, j > i \).

Proposition 2: If (for anhomogeneous traffics \( t(i,j) \)) a star network is optimal then the hub node \( p \) has the property that \( f^+_p + f^-_p = \max(f^+_i + f^-_i) \). That is, node \( p \) has the greatest sum of the traffics originating from and destined to it.

Proof: Suppose node \( p \) is selected as the hub node.

\[
Z(p) = \sum_{i \neq p} [t(p,i) + t(i,p)] + 2 \sum_{i \neq p, j \neq p} [t(i,j) + t(j,i)] \\
= 2 \sum_{i=1}^{n} \sum_{i=1}^{n} [t(i,j) + t(j,i)] - \sum_{i \neq p} [t(p,i) + t(i,p)] 
\]

Since the first term on the RHS is constant \( Z(p) \) is minimized by selecting \( p \) such that \( \sum_{i \neq p} [t(p,i) + t(i,p)] \) has the maximum possible value. The summation is equal to \( f^+_p + f^-_p \).

Since

\[
\sum_{j \neq p} [t(p,j) + t(j,p)] = \sum_{j \neq p} f^+_p = \sum_{j \neq p} f^-_p, 
\]

\[
f^+_p + f^-_p = \sum_{j=p+1}^{n} u_{jp} + \sum_{j=1}^{p} u_{jp} 
\]

(11)

and the hub node may be obtained directly from inspection of the upper triangular matrix \( U \).

Heuristic 2: Apply proposition 2 to find optimal star networks.

For the example with traffic matrix \( A \) the optimal star network also happens to be the optimal solution.

The motivation for including heuristic 2 in our set of heuristics is based on proposition 1. It is exact for homogeneous traffics and gives a compact structure. Nevertheless, it eliminates only \( n \) possible trees from the total \( n^{n-2} \).

6 Optimal Interconnection of Tree Networks

Consider interconnecting two arbitrary tree networks \( G_1[N_1; L_1] \) and \( G_2[N_2; L_2] \) by joining one node of \( G_1 \) to one node of \( G_2 \) with an additional link. If \( |N_1| = n_1 \) and \( |N_2| = n_2 \) then
there are \( n_1n_2 \) different ways of connecting the networks. Which two links should be selected in order to minimize the resultant total traffic carried on the links of the interconnected network, given additional O-D traffic requirements between nodes of \( G_1 \) and nodes of \( G_2 \)?

**Case 1: Homogeneous traffic**

Without loss of generality we may assume \( t(i, j) = 1 \) \( \forall (i, j) \).

**Proposition:** Given two trees \( G_1 \) and \( G_2 \) with mass functions \( Z_1 \) and \( Z_2 \) respectively, which are to be interconnected by a single link joining one node of \( G_1 \) to one node of \( G_2 \), the mass function of the interconnected network is minimized by connecting nodes with minimal mass functions.

**Proof:** Let \( i_0 \) and \( j_0 \) be arbitrary nodes in \( G_1 \) and \( G_2 \) respectively. Then the mass function \( Z(i_0, j_0) \) for the interconnected network \( G(i_0, j_0) \) is given by

\[
Z(i_0, j_0) = Z_1 + Z_2 + \sum_{i \in G_1, j \in G_2} [d(i, i_0) + d(j, j_0) + 1]
\]

where \( d(i, i_0) \) and \( d(j, j_0) \) are the distances from node \( i_0 \) to node \( i \) and node \( j_0 \) to node \( j \) respectively. The traffics weighting the distances from node \( i_0 \) are, in this case, incoming to node \( i_0 \) rather than outgoing.

\[
Z(i_0, j_0) = Z_1 + Z_2 + \sum_{i \in G_1, j \in G_2} [d(i, i_0) + d(j, j_0) + 1]f_{ij}
\]

and

\[
Z(i_0, j_0) = Z_1 + Z_2 + \sum_{i \in G_1, j \in G_2} [d(i, i_0) + d(j, j_0) + 1]t(i, j)
\]

which is minimized when \( Z_1 \) and \( Z_2 \) are least.

**Corollary:** If \( G_1 \) and \( G_2 \) are both star networks the optimal connection is achieved by joining the two hub nodes. This ability to treat the two network with a degree of independence when seeking an optimal interconnection will form the basis of the method proposed in section 7, but first we examine the above problem for the case of anhomogeneous offered traffics.

**Case 2: Anhomogeneous traffics**

Suppose that the two disjoint networks \( G_1[N_1; L_1] \) and \( G_2 [N_2; L_2] \) have external traffic requirements \( t(i, j) \) where \( i \in N_1 \Rightarrow j \in N_2 \) and \( i \in N_2 \Rightarrow j \in N_1 \). Let \( Z_1 \) and \( Z_2 \) be the traffic weighted mass functions for \( G_1 \) and \( G_2 \) respectively (these depend only on the internal traffics within \( G_1 \) and \( G_2 \)). We wish to identify two nodes \( i_0 \in N_1 \) and \( j_0 \in N_2 \) such that the traffic weighted mass functions for \( G[N; L] \), where \( N = N_1 \cup N_2 \) and \( L = L_1 \cup L_2 \cup \{(i_0, j_0)\} \) is minimized. We shall see that it is only necessary to consider the external traffics.

**Proposition 4:** For each node \( i \) in \( G_k(k = 1, 2) \) compute \( f^k_{ij} \) the sum of the external traffic flows from \( i \) to nodes in \( G_{3-k} \) and from nodes in \( G_{3-k} \) to \( G_k \). That is, the total external flows involving each node \( i \). Then, the optimal interconnection is \((i_0, j_0)\) where

\[
\sum_{i \in N_1} d(i, i_0)f^1_{ij} = \min_{k \in N_1} \sum_{i \in N_1} d(i, k)f^1_{ik}
\]

and

\[
\sum_{j \in N_2} d(j, j_0)f^2_{ij} = \min_{k \in N_2} \sum_{j \in N_2} d(j, k)f^2_{jk}
\]

**Proof:**

\[
Z(i_0, j_0) = Z_1 + Z_2 + \sum_{i \in N_1, j \in N_2} [d(i, i_0) + d(j, j_0) + 1]t(i, j)
\]

+ \sum_{j \in N_2} [d(j, j_0) + d(i, i_0) + 1]t(j, i)

\[
= Z_1 + Z_2 + \sum_{i \in N_1, j \in N_2} [d(i, i_0) + d(j, j_0) + 1]f_{ij}
\]

(5)

\[
= Z_1 + Z_2 + \sum_{j \in N_2} f_{ij} + \sum_{i \in N_1} d(i, i_0) \sum_{j \in N_2} f_{ij}
\]

+ \sum_{j \in N_2} d(j, j_0) \sum_{i \in N_1} f_{ij}

It is required then to minimize

\[
\sum_{i \in N_1} d(i, i_0)f^1_{ij} + \sum_{j \in N_2} d(j, j_0)f^2_{ij}
\]

where \( f^1_{ij} \) is the sum of all external traffic from node \( i \) in \( G_1 \) to all nodes in \( G_2 \) plus the sum of all external traffic from nodes in \( G_2 \) to node \( i \) in \( G_1 \). Similarly \( f^2_{ij} \) is the sum of all external traffics originating and terminating at node \( j \) in \( G_2 \). These sums can be called external incoming traffic weighted mass functions which we shall denote by \( \bar{Z}_i \) and \( \bar{Z}_j \) respectively. The traffics weighting the distances from node \( i_0 \) are, in this case, incoming to node \( i_0 \) rather than outgoing.

\[
Z(i_0, j_0) = Z_1 + Z_2 + \sum_{i \in N_1, j \in N_2} f_{ij} + \bar{Z}_i + \bar{Z}_j
\]

Since

\[
\sum_{i \in N_1, j \in N_2} f_{ij} \]

the minimisation is achieved by minimizing \( \bar{Z}_i \) and \( \bar{Z}_j \) independently.

7 A Strategy for Improving a Network

Proposition 4 shows us how to optimally join two tree networks. We shall apply this result in a different way. Starting from a tree structure (perhaps obtained as the 'best so far' using heuristics 1 and 2) we will remove a link, thereby forming two subtrees, and check whether optimal reconnection provides a new tree. If it does, it will do so at a lower, or the same, \( Z \) value.

To put the approach on a formal basis we first establish the following result

**Proposition 5:** Denote by \( \Delta Z(p_0, q_0; i_0, j_0) \) the change in the value of \( Z \) due to disconnecting link \((p_0, q_0)\) in \( G[N; L] \) to form \( G_1[N_1; L_1] \) and \( G_2[N_2; L_2] \) and subsequently connecting \( i_0 \in N_1 \) to \( j_0 \in N_2 \).

Then

\[
\Delta Z(p_0, q_0; i_0, j_0) = \bar{Z}_{i_0} - \bar{Z}_{j_0} - \bar{Z}_{p_0} - \bar{Z}_{q_0}
\]
Proof: Applying (15)
\[
\Delta Z(p_0, q_0; i_0, j_0) = (Z_1 + Z_2 + \sum_{i \in N_j} f_{ij} + \bar{Z}_{i_0} + \bar{Z}_{j_0})
- (Z_1 + Z_2 + \sum_{i \in N_i} f_{ji} + \bar{Z}_{p} + \bar{Z}_{q})
= \bar{Z}_{i_0} + \bar{Z}_{j_0} - \bar{Z}_{p_0} - \bar{Z}_{q_0}
\]

Decomposition and Optimal Interconnection Algorithm

Step 0 Apply the heuristics 1 and 2 and select the best approximate solution (arbitrary choice in case of ties)

Step 1 Consider deleting single links in turn. If \( \Delta Z \) computed from (16) is \( \leq 0 \) delete the link and rejoin according to the optimal interconnection result (proposition 4).

In case of ties multiple solutions may be found by following alternative computational ‘branches’.

Step 2 Stop when \( \Delta Z > 0 \) for all \( n - 1 \) links considered.

We note that the algorithm guarantees monotonic non-increasing values of \( Z \) with changing tree structures. Since the heuristics in Step 0 eliminate a large section of the feasible search region (giving a value of \( Z \) often within 10% of the lower bound), the Decomposition and Optimal Interconnection algorithm searches a relatively small section of the feasible region comprising the \( n^{n-2} \) tree structures.

Implementation of the Decomposition and Optimal Interconnection Algorithm is straightforward requiring simple direct numerical calculations based on \( t(i,j) \) and \( d(i,j) \). Updating distance is also straightforward.

Distance Updating
Suppose the deletion of a link \((p, q)\) results in two graphs \( G_1, G_2 \) and nodes in \( G_1 \) are \( i_0, i_1, i_2, \ldots \) and nodes in \( G_2 \) are \( j_0, j_1, j_2, \ldots \). The optimal interconnection is \((i_0, j_0)\).

\[
d(i_0, j_0) = 1
\]
\[
d(i_i, j_m) = d(i_0, i_0) + d(j_0, j_m) + 1
\]
other distances remain unchanged.

Does the algorithm always find \( Z^* \)?
This depends on whether it is possible to generate ‘local minima’. That is, structures for which an intermediate change to a greater value of \( Z \) subsequently leads to a lower value of \( Z \) than the current value. Trivially for \( n = 3 \) this is not possible. In a limited number of applications, the algorithm has always identified all optimal structures. This question is being considered.

8 Extensions of the Decomposition and Optimal Interconnection Method to Generalizations of Hu’s Problem

Some useful generalizations to the optimum requirement spanning tree problem are given in [7]. Whilst algorithms are given for these generalizations these all rely on the solution of maximal flow problems and the method of [2]. We are seeking alternative methods which are computationally faster and applicable to large networks.

The generalizations may be listed as follows:

1. restricting certain nodes to be ‘outer nodes’ (end nodes) of branches of the tree
2. specifying certain links as members of the final tree
   It is, for example, possible that there may be certain input or output devices which must be end nodes. It may also be necessary to retain some existing links in a communications network.
3. generating all optimal trees when the value \( t(i,j) \) for a specified pair of nodes varies in the interval \( [0, \infty) \).
4. including link costs (i.e. the distance between adjacent nodes is not necessarily 1). This gives an NP-complete problem.

To illustrate the decomposition and optimal interconnection algorithm for case 2 consider the 6 node example with matrix \( U \) below (only the total flows \( f_{ij} + f_{ji} \) are relevant). Suppose links \((2,5)\) and \((4,5)\) must remain fixed. Starting with the network below we consider the effect of deleting links \((2,6), (1,2)\) and \((2,3)\). The current value of \( Z \) is 81.

\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 \\
1 & 0 & 10 & 0 & 0 & 0 \\
2 & 0 & 4 & 0 & 0 & 3 \\
3 & 0 & 5 & 4 & 2 & 11 \\
4 & 0 & 7 & 2 & 0 & 0 \\
5 & 0 & 0 & 3 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
U =
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 \\
1 & 0 & 10 & 0 & 0 & 0 \\
2 & 0 & 4 & 0 & 0 & 3 \\
3 & 0 & 5 & 4 & 2 & 11 \\
4 & 0 & 7 & 2 & 0 & 0 \\
5 & 0 & 0 & 3 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
f^*_t
\]

\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 \\
1 & 0 & 10 & 0 & 0 & 0 \\
2 & 0 & 4 & 0 & 0 & 3 \\
3 & 0 & 5 & 4 & 2 & 11 \\
4 & 0 & 7 & 2 & 0 & 0 \\
5 & 0 & 0 & 3 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
f_t
\]

\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 \\
1 & 0 & 10 & 0 & 0 & 0 \\
2 & 0 & 4 & 0 & 0 & 3 \\
3 & 0 & 5 & 4 & 2 & 11 \\
4 & 0 & 7 & 2 & 0 & 0 \\
5 & 0 & 0 & 3 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

To illustrate the decomposition and optimal interconnection algorithm for case 2 consider the 6 node example with matrix \( U \) below (only the total flows \( f_{ij} + f_{ji} \) are relevant). Suppose links \((2,5)\) and \((4,5)\) must remain fixed. Starting with the network below we consider the effect of deleting links \((2,6), (1,2)\) and \((2,3)\). The current value of \( Z \) is 81.
Future studies will consider generalizations of these problems to other kinds of networks — in particular to network in which there are at least two disjoint paths between all nodes (or a specified subset).

This research has potential for application to structural design of future communications networks, for example MAN interconnection using virtual path network and ATM networks. The algorithms produced may also be useful in intelligent multicast routing networks.

Acknowledgement

I wish to thank Telecom Australia and the Australian Research Council for supporting this work.

References