Theory and Methodology

On design of a survivable network architecture for dynamic routing: Optimal solution strategy and an efficient heuristic

Iradj Ouveysi, Andrew Wirth *

Department of Mechanical and Manufacturing Engineering, The University of Melbourne, Parkville, Vic. 3052, Australia

Received 18 December 1996; accepted 24 November 1997

Abstract

We investigate network planning and design under volatile conditions of link failures and traffic overload. Our model is a non-simultaneous multi-commodity problem, with any particular two link failure being considered as one scenario. We show that the optimal solution model is not practically solvable for real-world problems and hence an efficient heuristic is provided which is $O(n^6)$ faster than the optimal model and is based on synthesizing a modified maximum spanning tree using an algorithm due to Gomory and Hu. The output of this procedure is then used to solve a much smaller linear program. Simulation results indicate that the heuristic is near optimal for problems with up to 40 nodes. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Heuristics; Linear programming; Survivability; Dynamic routing

1. Introduction

A communication network may be subject to link failure due to a catastrophic event or overload in link traffic during some specific times of the day or year. One way of dealing with such a problem is to establish a second network, in addition to the existing network, using the same nodes but introducing edges physically separate from the original network. We call this second network the protection network. The problem of finding the optimal protection network for one link failure or overload was solved by Gibbens and Kelly [1]. We address the problem of any two link failures or overloads in the original network. Using diverse protection links, our objective is to design and establish a protection network, such that in the case of any two link failures (or traffic overflow) in the original network, the protection network can handle this problem to achieve full service in the network. We assume the links in the protection network do not fail. Throughout this work we consider a fully meshed (or complete) network $G = (V, E)$ and a symmetric requirement matrix $r \in \mathbb{R}^{n \times n}$ ($r_{ij} = r_{ji} \geq 0$ and $|V| = n$, $|E| = m = \frac{n(n-1)}{2}$). By $e_{ij}$ we mean the edge between nodes $i$ and $j$. We

* Corresponding author. Tel.: +61 3 9344 4852; fax: +61 3 9347 8784.

1 An earlier version of this paper was presented at Globecom’95, Singapore.
also assume that the cost of assigning one unit capacity to any link in the network is equal to one.

Our objective is to construct a protection network \( \tilde{G} = (\tilde{V}, \tilde{E}) \) such that it has the capability of carrying traffic load as specified in the requirement matrix \( r \) for any two link failures of the network \( G \). The method can also be applied to deal with link congestion in \( G \).

The case of one link failure has been studied in [1–3]. Gomory and Hu [2] allow paths to contain many links (or edges) of the protection network but assume only one link of the original network has failed. They provide an exact algorithm for their problem. Following Gibbens and Kelly [1], we only allow one and two link paths but we extend their analysis to deal with two link failure. Two simple heuristics for this problem were presented in [4]. In this paper we use the Gomory and Hu procedure to construct a more accurate heuristic. Dynamic routing schemes such as Dynamically Controlled Routing [5], Dynamic Alternative Routing [6] or Dynamic Non-hierarchical Routing [7] could be applied to the augmented network, to handle traffic overload or link failures.

2. Optimal solution approach

We assume that the requirements due to link failure in the original network are to be satisfied by direct and two link paths in a protection network. By a failure scenario we mean the failure of two links in the original network. We define the following linear variables:

\( y_{ij} \) is the total amount of capacity assigned to a protection link \( e_{ij} \).

\( x_p^{ij} \) is the direct flow between nodes \( i \) and \( j \) along the protection link \( e_{ij} \) for failure scenario number \( p \).

\( x_{ijk}^p \) is the flow between nodes \( i \) and \( j \), through the tandem node \( k \) along the protection links \( e_{ik} \) and \( e_{kj} \) for scenario number \( p \).

Since a failure scenario corresponds to two links having failed, the total number of scenarios is

\[ \beta = \binom{m}{2}, \]

where \( m \) is the number of links of the complete network \( G \). Since

\[ m = \frac{n \times (n - 1)}{2}, \]

hence

\[ \beta = \binom{m}{2} = \frac{m!}{2!(m - 2)!} = \frac{(n^2 - 1)(n^2 - 2n)}{8}. \]

Now if we write the requirement constraints for any scenario, then the problem is similar to the non-simultaneous multi-commodity problem. For any scenario we need to satisfy some requirements for more than one pair of nodes, so the problem is multi-commodity and since at any instant we assume that only one scenario may occur, thus the problem is non-simultaneous.

Depending on whether the two failed links have a common end node or not we have the following two cases.

Case I: The two failed links have a common end node (see Fig. 1).

Assume that \( e_{ij} \) and \( e_{ik} \) are the two failed links of scenario number \( p \). Then the requirements between pairs \( (v_i, v_j) \) and \( (v_i, v_k) \) need to be satisfied by paths that use at most two hops in the protection network \( \tilde{G} \). The requirement constraints are:

(i) \( x_p^{ij} + \sum_{l \in V, l \neq i} x_{lij}^p \geq r_{ij} \)
for node pair \( (v_i, v_j) \),

(ii) \( x_p^{ik} + \sum_{l \in V, l \neq i} x_{ ilk}^p \geq r_{ik} \)
for node pair \( (v_i, v_k) \).

We also have the following capacity constraints:

(iii) \( y_{ij} - x_p^{ij} - x_{ijk}^p \geq 0, \)

Fig. 1. Two failed links have a common end node.
The total amount of capacity on any link needs to be at least equal to the amount of capacity assigned at any scenario. So we also need the following constraints:

(vii) \( y_{lj} - x_{q_{ij}}^l \geq 0 \quad \forall l \in V, \quad l \neq i, j, k, \)
(viii) \( y_{ik} - x_{q_{ik}}^l \geq 0 \quad \forall l \in V, \quad l \neq i, j, k. \)

Case II: The two failed links have no common end node (see Fig. 2).

Assume that \( e_{ij} \) and \( e_{kl} \) are the two failed links corresponding to scenario number \( q \). We need to satisfy the requirements between node pairs \( (v_i, v_j) \) and \( (v_k, v_l) \) via direct or two hop paths in the protection network. This implies that the following inequalities need to be satisfied:

(i) \( x_{q_{ij}}^l + \sum_{s \in \mathcal{V}, \ s \neq i} x_{q_{is}}^s \geq r_{ij} \)
for node pair \( (v_i, v_j) \),
(ii) \( x_{q_{kl}}^l + \sum_{s \in \mathcal{V}, \ s \neq k} x_{q_{skl}}^s \geq r_{kl} \)
for node pair \( (v_k, v_l) \).

The following capacity constraints are necessary.

(iii) \( y_{ij} - x_{q_{ij}}^l \geq 0 \)
(iv) \( y_{kl} - x_{q_{kl}}^l \geq 0 \).

For the commonly used links in above inequalities, we need to add following constraints:

(v) \( y_{ik} - x_{q_{ik}}^l \geq 0 \)
(vi) \( y_{ij} - x_{q_{ij}}^l \geq 0 \)
(vii) \( y_{kl} - x_{q_{kl}}^l \geq 0 \)
(viii) \( y_{lj} - x_{q_{lj}}^l \geq 0 \).

The total amount of capacity on any link needs to be at least equal to the amount of capacity assigned at any scenario. So we need the following constraints:

(ix) \( y_{ks} - x_{q_{ksl}}^l \geq 0, \quad s \in V, \quad s \neq i, j, k, l, \)
(x) \( y_{sl} - x_{q_{sij}}^l \geq 0, \quad s \in V, \quad s \neq i, j, k, l. \)

2.1. Number of constraints in each scenario

Here we calculate the number of constraints in both of the above mentioned cases. In Case I, in each scenario we have one constraint for each of (i)–(v). The number of constraints in (vi) is equal to \( n - 3 \) and number of constraints in each of the sets (vii) and (viii) is \( n - 3 \). So the total number of constraints in any scenario which corresponds to Case I is \( 3n - 4 \).

In Case II we have one constraint for any constraint set (i)–(viii). The number of constraints for each constraint set (ix)–(xii) is equal to \( (n - 4) \). So total number constraints in Case II is equal to \( 4n - 8 \).

2.2. Number of variables in each scenario

The number of \( x \) variables for any of the above mentioned cases is as follows: For Case I there are \( n - 1 \) \( x \) variables for each of constraint sets (i) and (ii). Hence total number of \( x \) variables for each scenario in Case I is \( 2n - 2 \). Obviously the number of \( x \) variables for Case II is also equal to \( 2n - 2 \).

2.3. The total number of variables and constraints

The number of scenarios of Case I is \( \beta_1 = n(n-1)(n-2)/2 \), since there are \( n \) ways of choosing \( i \), the \( n - 1 \) ways of choosing \( j \) and \( n - 2 \) ways of choosing \( k \) as shown in Fig. 1. Further
The total number of scenarios of Case II is equal to:
\[ \beta_{II} = \beta - \beta_1, \]
\[ \beta_{II} = \frac{n(n - 1)(n - 2)(n - 3)}{8}. \]

Now it is easy to calculate the total number of pivots for this problem. Assume that, as in most general purpose linear programming packages, we use the simplex method. Published bounds on the average number of pivots is \( O(\min \{m^2, n^2\}) \). In any event, using the \( O(\min(m^2, n^2)) \) result would only improve the relative performance of the heuristic, to be discussed shortly, when compared to the optimal solution approach, as far as computational complexity is concerned.

The complexity of one pivot: Assume that the simplex tableau at any given iteration is as follows:

\[
\begin{array}{c|c}
D & f \\
\hline
u & \gamma
\end{array}
\]

where \( D = (d_i) \) is an \( \bar{m} \times (\bar{n} + \bar{m}) \) matrix, \( f = (\varphi_1, \ldots, \varphi_m)^T \) is a column \( m \)-vector, \( u = (v_1, \ldots, v_{n+m}) \) is a row \( (\bar{n} + \bar{m}) \)-vector (reduced cost vector), and \( \gamma \) is a real number. In performing a pivot step, we pick a pivot element, say \( d_{ik} \), and perform row operations so as to obtain in the \( k \)th column of the tableau only zeroes except for the 1 in the \( l \)th position. The complexity of these row operations is \( O(\bar{m} \times (\bar{n} + \bar{m})) \), as we need to modify all entries of the simplex tableau given above.

Table 1 shows the complexity of the optimal solution approach for survivable network design, where two links are subject to failure simultaneously.

### 3. Complexity analysis

In performing the complexity analysis, we assume that, as in most general purpose linear programming packages, we use the simplex method. We first cite results on the average number of pivots for solving a linear programming problem and then examine the complexity of performing one pivot.

The average number of pivots: The best established bound on the average number of pivots is \( O(\min(n^2, \bar{n}^2)) \) (see for example, [8]), where \( \bar{m} \) is the number of constraints and \( \bar{n} \) is the number of variables. However in practice it has been found that the number of pivot steps is about linear in the problem dimensions (see for example [9], p. 139). This is the bound we use here (namely \( O(\min(\bar{m}, \bar{n})) \)). In any event, using the \( O(\min(m^2, n^2)) \) result would only improve the relative performance of the heuristic, to be discussed shortly, when compared to the optimal solution approach, as far as computational complexity is concerned.

### 4. Heuristic solution

**Definition 1.** Given a graph \( G = (V, E) \) and a subset of nodes \( \omega \subset V \), we denote by \( G[\omega] \) the...
subgraph of $G$ induced by $\omega$ and by $E[\omega]$ its edge set $\{e_{ij} \in E \mid i, j \in \omega\}$.

**Definition 2.** Let $\tilde{G} = (V, \tilde{E})$ be the protection network for a given network $G = (V, E)$. Let $c_{ij}$ denote the capacity of the edge $e_{ij} \in \tilde{G}$. We assume that the requirements due to link failure are satisfied by one and two link paths. Hence there is $x_{ij}$ capacity between two nodes $i$ and $j$ in $\tilde{G}$, where

$$\sum_{l \in V, j \neq i, l} c_{lj} + c_{ij} = x_{ij}$$

and $c_{ij}$ is the amount of capacity of the 2-link path $\{i, c_{il}, l, e_{lj}, j\}$, defined as

$$c_{ij} = \min\{c_{il}, c_{lj}\}.$$

### 4.1. Preliminaries

Before we describe the construction of the protection network, we give the following definitions for minimum and maximum spanning trees.

**Definition 3** (Minimum spanning tree). Given a connected undirected graph $G = (V, E)$ (where $V$ is the set of nodes and $E$ is the set of edges of the graph) and a length function $l : E \to \mathbb{R}$, we say $T_{\text{min}} = (V, E_{\text{min}})$ is the minimum spanning tree of the graph $G$, if $T_{\text{min}}$ is connected, has no cycle and has the minimum total length.

**Definition 4** (Maximum spanning tree). Given a connected undirected graph $G = (V, E)$ (where $V$ is the set of nodes and $E$ is the set of edges of the graph) and a length function $l : E \to \mathbb{R}$, we say $T = (V, E_T)$ is the maximum spanning tree of the graph $G$, if $T$ is connected, has no cycle and has the maximum total length.

The problem of finding the minimum spanning tree (and hence the maximum spanning tree) is polynomially solvable, e.g., by Prim’s algorithm [10]. The concept of maximum spanning tree will be used in the synthesis of a protection network.

### 4.2. Network synthesis algorithm

If we allow for one link failure only, it is clear that the maximal spanning tree will be an adequate, but non-optimal protection network. Gomory and Hu [2] found the optimal solution for links of arbitrary length. The following procedure due to Gibbens and Kelly [1] modifies this for direct and two link paths.

**Step 1.** Let $T_n$ be the maximum spanning tree and assume that $S$ is the set of edges of $T_n$ and the values on the edges of $T_n$ represent the requirements. Set $i = 1$;

**Step 2.** Find an edge $e_{st}$ from $S$ with minimum value, set $\hat{r}_i = r_{st}$. Then using other edges in $S$ find a maximal connected uniform tree $t_i$, with all edges having the same requirement as $e_{st}$;

**Step 3.** In the original tree $T_n$ reduce the requirement of each edge which exists in $t_i$ by value of requirement of $e_{st}$;

**Step 4.** Delete from $S$ any edge which has the requirement value of zero;

**Step 5.** If $S$ is empty, go to step 6, otherwise set $i = i + 1$ and go to step 2;

**Step 6.** Synthesize any uniform tree $t_i = (V', E_n)$ by providing $\beta'$ capacities on all links in $E[V']$, where

$$\beta' = \frac{\hat{r}_i}{|V'| - 1}.$$

Let us denote the resultant fully meshed network as $g_{\hat{r}_i}$, where $g_{\hat{r}_i} = (V', E'[V'])$;

**Step 7.** Adding up all $g_{\hat{r}_i}$’s will provide $\tilde{G}$, stop.

### 4.3. Protection network

Since we are dealing with two link instead of one link failure, we need to modify the maximum spanning tree before synthesizing it. We use the following procedure to modify the requirement values of the links of the maximum spanning tree.

**Procedure A:**

**Step 1.** Let $S_1 = E_m$ and $S_2 = E \setminus E_m$. Set $i = 1$;

**Step 2.** Choose a link $e_{st}$ arbitrarily from $S_1$ and modify its requirement as follows:
\[
\bar{r}_u = \max\{\bar{r}_u^1, \bar{r}_u^2\},
\]

where

\[
\bar{r}_u^1 = \max\{r_{st} + r_{us} | e_{us} \in S_2\text{ and adding it to } T_m \text{ makes a cycle which contains } e_{us}\}
\]

\[
\bar{r}_u^2 = \max\{r_{st} + r_{us} | e_{us} \in S_2, u = s, t \text{ and } v \in V, v \neq s, t\}
\]

Step 3. Delete \(e_{st}\) from \(S_1\);

Step 4. If \(i = n - 1\) go to step 5, otherwise set \(i = i + 1\) and go to step 2;

Step 5. Stop.

4.4. The main algorithm

Step 1. Given \(G = (V, E)\) and requirement matrix \((r_{ij})\), construct the maximum spanning tree \(T_m\).

Step 2. Modify the requirements of links in \(T_m\) by Procedure A.

Step 3. Synthesize the modified maximum spanning tree by the Gomory–Hu Algorithm mentioned above. Let \(\tilde{G} = (V, \tilde{E})\) be the resultant network obtained from the synthesis procedure and assume \((\tilde{e}_{ij})\) represent the capacities on the links of \(\tilde{G}\). Now \(\tilde{G}\) can handle any two link failures unless both are from \(E_m\), as is proved in Theorems 1 and 2 below.

Step 4. Apply Procedure B (see Section 4.6) with \(\tilde{G}\) as input. Let \(\tilde{G}\) be the output of Procedure B. The protection network \(\tilde{G}\) can handle any two link failures of the original graph \(G\).

Step 5. Stop.

4.5. An example

As an example we consider the network in Fig. 3, where the numbers on the links represent the end-to-end requirements. The steps for Procedure A and the synthesis process are shown in Fig. 3.

The following sequence of lemmas and theorems shows that the synthesized modified maximum spanning tree allows for the failure of any two links, provided only that at least one is not in the maximum spanning tree (MST). To allow for the case where both failure links are in the MST, we must solve a linear program (Procedure B, Section 4.6).

Lemma 1. Consider two links \(e_{ij}\) and \(e_{jk}\) in \(E_m\) with modified requirement values \(\bar{r}_{ij}\) and \(\bar{r}_{jk}\). Assume, without loss of generality that \(\bar{r}_{ij} \leq \bar{r}_{jk}\), then there is at least \(\bar{r}_{ij}\) capacity in \(\tilde{G}\) between nodes \(i\) and \(k\) (see Fig. 4).

Proof. If \(\bar{r}_{ij} \leq \bar{r}_{st} \forall e_{st} \in E_m\), then by the synthesis process the value \(\bar{r}_{ij}\) is uniformly synthesized in the whole of the network \(E[V]\) and hence there would be \(\bar{r}_{ij}\) capacity between nodes \(i\) and \(k\) in \(G\). If \(\bar{r}_{ij} \not\leq \bar{r}_{st} \forall e_{st} \in E_m\), then let \(e_{si,t1} \in E_m\) be a link, such that \(\bar{r}_{si,t1} \leq \bar{r}_{st} \in E_m\). Then by the synthesis process the value \(\bar{r}_{si,t1}\) is uniformly synthesized between edges of \(E[\omega^1]\), where \(\omega^1 = V\). In this way a capacity equal to \(\beta^1 = \bar{r}_{si,t1}/(|\omega^1| - 1)\) is assigned for all links in \(E[\omega^1]\). Recall from the synthesis algorithm that at this stage \(e_{si,t1}\) and all other links which have \(\bar{r}\) value equal to \(\bar{r}_{si,t1}\) are deleted from \(T_m\). Let the resultant disconnected subgraphs be denoted by \(T_{m1}, T_{m2}, \ldots, T_{mp}\). Obviously \(e_{ij}\) and \(e_{jk}\) both belong to a subgraph, say \(T_{m1} = (V_{m1}, E_{m1})\). In \(T_{m1}\) any link \(e_{st}\) has the new requirement value equal to \(\tilde{r}_{st} = \bar{r}_{st} - \bar{r}_{si,t1}\). Now either \(\tilde{r}_{st}^{(1)}\) is the minimum among all new link values in \(T_{m1}\) and is uniformly synthesized between \(E[V_{m1}]\) which together with step 1 provides

\[
\bar{r}_{si,t1} + \tilde{r}_{st}^{(1)} = \bar{r}_{si,t1} + \bar{r}_{st} - \bar{r}_{si,t1} = \bar{r}_{ij}
\]

and we are done, or there is a link, say \(e_{st2} \in E_{m1}\) which has the smallest \(\tilde{r}\) value between all links in \(E_{m1}\). In the latter case a new capacity equal to \(\beta^2 = \tilde{r}_{st2}/(|V_{m1}| - 1)\) is assigned to the links in \(E[V_{m1}]\) and the requirement of any link in \(E_{m1}\) is updated as follows:

\[
\tilde{r}_{st}^{(2)} = \tilde{r}_{st}^{(1)} - \tilde{r}_{st2}.
\]

By doing these steps, finally we reach a step \(q\) in which the updated requirement value on link \(e_{ij}\) is
the smallest among all links in the subgraph containing link $e_{ij}$ at this step, say $T_{mf} = \langle V_{mf}, E_{mf} \rangle$. Then we synthesize the remaining requirement value on $e_{ij}$, say $r_{ij}^{(q-1)}$ between all links of $E[V_{mf}]$, where $V_{mf} \supseteq \{i, j, k\}$. Hence the existing capacity between nodes $i$ and $k$ is equal to

Fig. 3. An example of the application of the main algorithm.
Corollary 1. Let \( P = \{i, e_{ij}, j, \ldots, e_{kl}, l\} \) be a path in \( T_m \) and assume \( e_{st} \) is the link in \( P \) with minimum modified requirement \( \bar{r}_{st} \). Then \( \alpha_{st} \geq \bar{r}_{st} \).

**Proof.** By using the same logic as the proof of Lemma 1 it is easy to show that a value equal to \( \bar{r}_{st} \) is uniformly synthesized among \( E[\omega] \), where any node \( k \) of \( P \) is a member of \( \omega \) also.

Corollary 2. For any two links \( e_{ij} \in E \setminus E_m \) and \( e_{ik} \in E_m \), there is at least \( r_{ij} + r_{ik} \) requirement between nodes \( i \) and \( j \) in modified \( T_m \).

**Lemma 2.** In the protection network \( \tilde{G} \), for any link \( e_{ij} \in E \setminus E_m \), there is \( \alpha_{ij} \geq 2r_{ij} \) capacity between two nodes \( i \) and \( j \). Besides \( \alpha_{ij} \) is uniformly synthesized between \( i \) and \( j \) and some other node (nodes) of \( G \).

**Proof.** Adding \( e_{ij} \) to \( T_m \) makes a cycle, say \( Q \). In \( Q \) any link \( e_{st} \) has requirement at least equal to \( r_{st} \) which after the modifying step increases to \( r_{st} + r_{ij} \). Then by Corollary 1, there is at least \( 2r_{st} \) capacity between nodes \( i \) and \( j \) in \( \tilde{G} \). So we have \( \alpha_{ij} \geq 2r_{ij} \) (see Fig. 5).

Since all links in \( Q \) (except \( e_{ij} \)) have \( \bar{r} \) value at least equal to \( 2r_{ij} \), then \( \alpha_{ij} \) is uniformly synthesized between \( E[\omega] \). Where \( \omega \) is the set of nodes of \( Q \).

**Theorem 1.** Let \( \tilde{G} \) be the protection network obtained by executing the network synthesis algorithm on the modified maximum spanning tree of the network \( G \). Then \( \tilde{G} \) has sufficient capacity to carry the traffic requirements of any two failed links from \( E \setminus E_m \).

**Proof.** We distinguish two cases.

**Case a:** Two failed links have a common end node (see Fig. 6).

Let \( e_{ij}, e_{ik} \in E \setminus E_m \) be two failed links. By Lemma 2 there is at least \( 2r_{ik} \) capacity in \( \tilde{G} \) between \( i \) and \( j \). Also by the same reason there is at least \( 2r_{ik} \) capacity between \( i \) and \( k \) in \( \tilde{G} \). So we have:

\[
\alpha_{ij} \geq 2r_{ij},
\]

\[
\alpha_{ik} \geq 2r_{ik}.
\]

Let \( r_{ij} \geq r_{ik} \). We need to show that the commonly used links have enough capacity to satisfy the requirements \( r_{ij} \) and \( r_{ik} \). Here commonly used links are \( e_{ij} \) and \( e_{il} \), \( l \in V, \ l \neq i \).

**Fig. 5.** For any \( e_{ij} \in E \setminus E_m \) we have \( \alpha_{ij} \geq 2r_{ij} \).
Let $c_{ilj}$ represent the minimum capacity on the protection links $eil$ and $elj$. From inequalities (1) and (2) we have:

$$c_{ij} + \sum_{l \in V, l \neq i,j} c_{ilj} \geq 2r_{ij},$$

$$c_{ik} + \sum_{l \in V, l \neq i,k} c_{ilk} \geq 2r_{ik}. \quad (3)$$

Inequality (3) implies that using half of the capacities of the common used links satisfies the requirements of $r_{ij}$ as

$$\frac{1}{2} c_{ij} + \frac{1}{2} \sum_{l \in V, l \neq i,j} c_{ilj} \geq r_{ij}. \quad (4)$$

By providing $r_{ij}$, in the worst case we use half of the capacities of the commonly used links (we say the worst case because $\min\{c_{il}, c_{ilj}\}$ is not necessarily $c_{il}$). Now if $\min\{c_{il}, c_{ilj}\} = c_{il}$ then there is $\frac{1}{2} c_{il}$ capacity unused on protection link $eil$ in $G$. That means we have $\frac{1}{2} c_{il}$ spare capacity on the two-hop path $\{i, e_{il}, l, e_{lk}, k\}$. If $\min\{c_{il}, c_{ilk}\} = c_{ilk}$, then $\frac{1}{2} c_{ilk}$ and this shows that we have $\frac{1}{2} c_{ilk}$ capacity of two-hop path $\{i, e_{il}, l, e_{lk}, k\}, \forall l \in V, l \neq i$.

Hence the capacity equal to

$$\frac{1}{2} c_{ik} + \frac{1}{2} \sum_{l \in V, l \neq i,k} c_{ilk}$$

is available for $r_{ik}$ and from Eq. (4) we know that

$$\frac{1}{2} c_{ik} + \frac{1}{2} \sum_{l \in V, l \neq i,k} c_{ilk} \geq r_{ik}. \quad (4)$$

**Case b**: Two failed links have no common end node.

Let $e_{ij}$ and $e_{kl}$ be two failed links. Adding $e_{ij}$ to $T_m$ makes a cycle, say $Q^1 = (V^{\oplus}, E^{\oplus})$ in $T_m$ and adding $e_{kl}$ to $T_m$ makes a cycle in $T_m$, say $Q^2 = (V^{\oplus}, E^{\oplus})$.

Let

$$\bar{r}_{x'x} = \min_{e_{x'x} \in E^{\oplus}} \bar{r}_{x'x}, \quad \bar{r}_{y'y} = \min_{e_{y'y} \in E^{\oplus}} \bar{r}_{y'y}. \quad (4)$$

By Lemma 2 we have:

$$\bar{r}_{x'x} \geq 2r_{ij}, \quad \bar{r}_{y'y} \geq 2r_{kl}. \quad (4)$$

Assume, without loss of generality that $\bar{r}_{x'x} \leq \bar{r}_{y'y}$. Now we consider the following two sub-cases.

**Sub-case i**: $Q^1$ and $Q^2$ have some common nodes (see Fig. 7). Here by the same logic as the
proof of Lemma 1 it is easy to show that up to some step, say $q^1 \geq 1$, $r_{i'i'}$ is uniformly synthesized among $E[\omega^1]$, where $\omega^1 \supseteq \{V^0 \cup V^0\}$. Using half of the capacities of the links $e_{ij}$ and $e_{i'j'}$, $l' \in \omega^1$, $l' \neq i,j$, provides the requirements $r_{ij}$ for node pair $(i,j)$. In the following steps, say up to step $q^2$ a value equal to $r_{i'i'}$ is uniformly synthesized among $E[\omega^2]$, where $\omega^2 \supseteq V^0$. Using half of the capacities which are assigned up to step $q^1$, of the links $e_{kl}$ and $e_{k'l'}$, $l' \in \omega^1$, $l' \neq k,l$, provides $\eta_1 = r_{i'i'}/2$ capacity for node pair $(k,l)$ and by using half of the capacities which are assigned through steps $q^1 + 1$ up to $q^2$, of the following links: $e_{kl}$ and $e_{k'l'}$, $l' \in \omega^2$, $l' \neq k,l$, we achieve $\eta_2 = (r_{i'i'} - r_{k'l'})/2$ more capacity for node pair $(k,l)$. Hence in total we have

$$\eta_1 + \eta_2 = \frac{r_{i'i'} - r_{k'l'}}{2} = \frac{r_{i'i'}}{2} \geq r_{kl}.$$

Sub-case ii: $Q^1$ and $Q^2$ have no common node (see Fig. 8).

Let $P = (V^p, E^p)$ be the path $T_m$ that connects $Q^1$ to $Q^2$ and let $e_{p lp}$ be a link in path $P$ which has the smallest $r$ value among all links in this path. If $r_{p lp} \leq r_{i'i'}$ then at some step of the synthesis process, once $r_{p lp}$ is uniformly synthesized among $E[V^0 \cup V^0 \cup V^p]$, then $Q^1$ and $Q^2$ will be disconnected. Using half of the capacities assigned up to this stage, of the links in $E[V^0 \cup V^0 \cup V^p]$ will provide $r_{p lp}$ capacity for node pair $(i,j)$ or $(k,l)$, or $\eta_1 = r_{p lp}/2$ capacity for node pairs $(i,j)$ and $(k,l)$. After this step since the two cycles $Q^1$ and $Q^2$ have been disconnected, so up to some stage $q^2$, $r_{i'i'} - r_{p lp}$ is uniformly synthesized in $E[\omega^1]$ where

$$\omega^1 \supseteq V^0, \quad \omega^1 \cap V^0 = \emptyset.$$

Using half of the capacities of the links in $E[\omega^1]$ which are assigned in the steps $q^1 + 1$ up to $q^2$, we can satisfy $\eta^1 = (r_{i'i'} - r_{p lp})/2$ more capacity for node pair $(i,j)$ and we have

$$\eta_1 + \eta^1 = \eta_1 + \frac{r_{i'i'} - r_{p lp}}{2} = \frac{r_{i'i'}}{2} \geq r_{ij}.\]

Hence there is enough capacity to satisfy $r_{ij}$ for node pair $(i,j)$. Similarly there is enough capacity to satisfy $r_{kl}$ for node pair $(k,l)$.

If $r_{p lp} > r_{i'i'}$ but $r_{p lp} \leq r_{k'l'}$, then the argument as in the proof of Lemma 1 implies that $G$ is feasible in this case also. For the case that $r_{p lp} > r_{i'i'}$, the proof is the same as for Sub-case i.

**Theorem 2.** Let $G$ be the protection network obtained by executing the network synthesis algorithm on the modified maximum spanning tree of the network $G$. Then $G$ has sufficient capacity to carry the traffic requirements of any two failed links one from $E_m$ and another from $E \setminus E_m$.

**Proof.** Here we distinguish the following two cases.

**Case a:** Two failed links, $e_{ij} \in E_m$ and $e_{ik} \in E \setminus E_m$ have a common end node (see Fig. 9). Adding $e_{ik}$ to $T_m$ makes a cycle, say $Q = (V^0, E^0)$.

Sub-case i: If $e_{ij} \in E^0$, then obviously $r_{at} \geq r_{ik}$, $\forall e_{at} \in E^0$ and hence any link $e_{at} \in E^0$ has at least a modified requirement value equal to $r_{at} + r_{ik}$.

![Fig. 8. $Q^1$ and $Q^2$ have no common node.](image)
Let $a_1 \hat{=} \min f_{\text{rst}} \hat{\in} \hat{E} \setminus \hat{E}_m$.

Then $a_1$ is uniformly synthesized between edges in $E[\omega^1]$, where $\omega^1$ is a subset of $V$ containing $\{i, j, k\}$. The remaining demand $(a_{ij} - a_1)$ is synthesized between $E[\omega^2]$, where $\omega^2 \subset \omega^1$. Let $\rho_1 = a_{ij}/a_1$, then for node pair $(i, k)$ we have

$$\rho_1 c_{ik} + \rho_1 \sum_{l \in \omega^1, l \neq i, k} c_{ilk} \geq r_{ik}.$$ 

For node pair $(i, j)$ from the former step of synthesizing we have

$$\eta_1 = (1 - \rho_1) a_1 = \frac{a_{ij} - r_{ik}}{a_1} \times a_1 = a_{ij} - r_{ik}.$$ 

In the following steps of the synthesis algorithm, we provide $\eta_2 = a_{ij} - a_1$ more capacity between $i$ and $j$. Hence in total we have

$$\eta_1 + \eta_2 = (a_{ij} - r_{ik}) + (a_{ij} - a_1) = a_{ij} - r_{ik} \geq (r_{ij} + r_{ik}) - r_{ik} = r_{ij}.$$ 

Sub-case ii: If $e_{ij} \notin \hat{E}_m$ then by Corollary 2 we have $r_{ij} \geq r_{ij} + r_{ik}$, and with the same logic as Sub-case i, it is easy to show that $G$ is feasible if $e_{ij}$ and $e_k$ are failed.

Case b: Two failed links, $e_{ij} \in \hat{E}_m$ and $e_{kl} \in \hat{E}_m \setminus \hat{E}_m$ have no common end node (see Fig. 10). Here we distinguish two sub-cases.

Sub-case iii: Adding $e_{kl}$ to $T_m$ makes a cycle, say $Q = \langle V^G, E^G \rangle$ which contains $e_{ij}$. Then the proof of this sub-case is exactly the same as the proof of Sub-case i.

Sub-case iv: $e_{kl}$ makes a cycle in $T_m$, say $Q = \langle V^G, E^G \rangle$, where $e_{ij} \notin \hat{E}_m$.

Let $P = \langle V^P, E^P \rangle$ be the path in $T_m$ connecting $Q$ to $e_{ij}$. Without loss of generality we assume that $e_{ij}$ has the maximum $r$ value among all links of $E_m \setminus E'$ and let $x_1 = \min \{r_{st} | e_{st} \in E \setminus E_m \}$. If $x_1 \geq 2r_{ij}$

![Fig. 9. Two failed links have a common end node.](image9.png)

![Fig. 10. Two failed links have no common end node.](image10.png)
then since $x_1 \geq 2r_{kl}$ (by Corollary 2) then the problem is trivial. Let us define $x_2 = \min\{\bar{r}_{ij} \mid e_{ij} \in E^p\}$. Now if $x_1 < 2r_{ij}$ we have the following two scenarios.

Scenario I: $x_1 < x_2$. Here at first step $x_1$ is uniformly synthesized between $E[\omega^1]$ where $\omega^1 = V$ and we assign $\beta^1 = x_1/(|V| - 1)$ capacity on all links. Hence for node pair $(k, l)$ we have

$$c_{kl} + \sum_{l \neq k, i, j} c_{kl'} = \beta^1 + (n - 4)\beta^1$$

$$= (n - 3) \times \frac{x_1}{n - 1} \geq \frac{x_1}{2} \text{ since } n \geq 5.\quad (5)$$

Since $z_1 \geq 2r_{kl}$, then the right-hand side of inequality (5) is greater than or equal to $r_{kl}$. Hence there is enough capacity between $k$ and $l$ to satisfy the requirement $r_{kl}$ without using any capacity of the common links $e_{kl}, e_{ij}, e_{il}$ and $e_{ij}$ (these are the links which are used commonly to satisfy some of the requirements of the failed links).

For node pair $(i, j)$ since there is capacity equal to

$$c_{ij} + \sum_{l \in V} c_{ij} = \bar{r}_{ij} \geq r_{ij}$$

provided by the synthesis algorithm, so there is enough capacity in the protection network to handle this case of the failure scenario.

Scenario II: $x_1 \geq x_2$. Here at least $x_1$ capacity is assigned between nodes $V^0$, that is we have

$$c_{kl} + \sum_{l \in V^0, l \neq k, i, j} c_{kl'} \geq \frac{x_1}{2} \geq r_{kl}.$$  

By the above inequality we showed that without using the common links (mentioned above) we can provide the requirement for node pair $(k, l)$. Now since $\bar{r}_{ij}$ capacity is available between nodes $i$ and $j$ (the total capacity of direct and 2-link paths), then we have

$$c_{ij} + \sum_{l \in V} c_{ij} \geq \bar{r}_{ij} \geq r_{ij}.$$  

By the above statements we proved that our protection network $\tilde{G}$ can provide full protection in case of two link failures, one from $E_m$ and one from $E \setminus E_m$.

4.6. Construction of $\tilde{G}$ from $\tilde{G}$

Here we use a linear programming approach to construct $\tilde{G}$ from $\tilde{G}$. Recall from Theorems 1 and 2 that, $\tilde{G}$ provides full protection of the original network in the case of any two link failures unless both are from $E_m$, now we consider the case when we have two link failures from $E_m$. Since $|E_m| = n - 1$, so the total number of failure scenarios is equal to

$$\text{TNS} = (|E_m|) = \left( \frac{n - 1}{2} \right) = \frac{(n - 1)(n - 2)}{2}.$$  

From Sections 2.1 and 2.2 it follows that $\text{TNS} = O(n \times \text{TNS})$ and $\text{TNC} = O(n \times \text{TNS})$.

**Procedure B:**

Solve the linear program

$$\min \sum_{j=1}^{n} \sum_{i=1}^{n} y_{ij} \quad \text{(HLP)}$$

subject to the above TNC set of constraints and $y_{ij} \geq \bar{e}_{ij} \forall e_{ij} \in \hat{E}$ (total number of this type constraint is of the order of $n^2$ and hence the order of TNC does not change by adding these constraints). Here $x$ and $y$ variables are non-negative variables and $\bar{c}_{ij}$ is the amount of capacity assigned to any link $e_{ij} \in \hat{E}$ at the end of the synthesis process.

Here solving the linear program model (non-simultaneous multi-commodity problem) is easier because by using the mentioned heuristic we reduce the problem size and apply the linear programming model for the case where the number of scenarios is equal to $|E_m| - 1$.

4.7. Complexity of the heuristic

It is easy to verify that the most expensive part of the heuristic is the LP part. The optimal solution strategy in Section 2 requires the solution of a large scale linear program. The heuristic allows for a major decrease in the size of the linear program and a corresponding decrease in computational...
complexity. Table 2 summarizes the complexity of the heuristic based on our earlier discussions.

### 4.8. A lower bound

The following lemma provides a lower bound for our problem.

**Lemma 3.** For a given network \( G = (V, E) \) with a symmetric non-negative requirement matrix \( r \in \mathbb{Z}_{+}^{n \times n} \), let \((c^*_ij)\) represent an optimal solution, then

\[
\sum_{i<j} c^*_ij \geq \sum_i \left( \frac{\hat{r}^{(1)}_i + \hat{r}^{(2)}_i}{2} \right),
\]

where \( \hat{r}^{(1)}_i \) and \( \hat{r}^{(2)}_i \) denote the largest two traffic requirements emanating from node \( i \).

The total outgoing (or incoming) capacity at each node in an optimal protection network exceeds or is equal to the sum of the capacity requirements of any two links ending at that node (this is a necessary condition for protection against any two link failure). Using this fact we have

\[
\sum_{j \in V, j \neq i} c^*_ij \geq \hat{r}^{(1)}_i + \hat{r}^{(2)}_i.
\]

Summing the above inequalities over all nodes of \( V \), we obtain

\[
\sum_{j \neq i} \sum_{i} c^*_ij \geq \sum_i \left( \hat{r}^{(1)}_i + \hat{r}^{(2)}_i \right).
\]

Since \( c^*_ij = c^*_ji \), the above inequality can be simplified to

\[
\sum_{i<j} c^*_ij \geq \sum_i \left( \frac{\hat{r}^{(1)}_i + \hat{r}^{(2)}_i}{2} \right).
\]

### 5. Implementation of heuristic

We checked the effectiveness of the heuristic by solving some randomly generated problems. Whenever possible \((n \leq 10)\) we compared our results with the optimal solution, using CPLEX optimizer package (Linear Optimizer 4.0.9) [11] for the model in Section 2. For larger \( n \) we compared the heuristic’s performance with the lower bound value.

Define

\[
\eta = \frac{Z^h - Z^{op}}{Z^{op}} \quad \text{or} \quad \eta = \frac{Z^h - Z^{lb}}{Z^{lb}},
\]

where \( Z^{op} \) is the optimal solution of the large scale optimization model of the problem, \( Z^h \) the solution which is found by the heuristic, \( Z^{lb} \) the lower bound as defined in Lemma 3 (we use this if \( Z^{op} \) is not available).

The summary of the implementation results is given in Table 3. Here \( n \) is the number of the nodes in the network and \( m \) the number of links in the network, \( t_{op} \) is the running time (CPU seconds on a SPARC station 2) of the optimizer package CPLEX to solve the large scale optimization model of the problem to achieve the optimal solution and \( t_{h} \) is the running time of the optimizer package CPLEX to solve the last step of the heuristic in CPU seconds. It is seen that the solution time by the LP optimizer package is decreased by more than 1000 times for problem with \( n = 10 \) when the heuristic is used.

### 6. Conclusions

The problem of finding the optimal protection network, for a given fully meshed network, which allows for two link failures has been shown to lead to a large scale linear program. The application and extension of a network synthesis algorithm, due to Gomory and Hu and Gibbens

<table>
<thead>
<tr>
<th>Complexity analysis of the heuristic</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of LP problem to be solved</td>
<td>HLP</td>
</tr>
<tr>
<td>Number of LP problems to be solved</td>
<td>1</td>
</tr>
<tr>
<td>Number of constraints of the LP</td>
<td>( O(n^3) )</td>
</tr>
<tr>
<td>Number of variables of the LP</td>
<td>( O(n^3) )</td>
</tr>
<tr>
<td>Number of simplex pivots for the LP</td>
<td>( O(n^3) )</td>
</tr>
<tr>
<td>Complexity of one simplex pivot for the LP</td>
<td>( O(n^3) )</td>
</tr>
<tr>
<td>Overall complexity of the method</td>
<td>( O(n^3) )</td>
</tr>
</tbody>
</table>
and Kelly, which gives the optimal solution for one link failure, resulted in a drastic reduction in the size of the linear program. The resulting heuristic has been shown to perform well, at least for the moderately sized problems on which it was tested.

### References


[8] M.J. Todd, Polynomial expected behaviour of a pivoting algorithm for linear complementarity and linear program-

---

### Table 3
The implementation results of the heuristic compared with the optimal solutions

<table>
<thead>
<tr>
<th>Problem indicators</th>
<th>Lower bound $Z^\text{lb}$</th>
<th>Optimal solution approach $Z^\text{op}$</th>
<th>Results obtained by applying heuristic $Z^\text{h}$</th>
<th>$\eta%$</th>
<th>$\delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$ $m$ $\beta$</td>
<td></td>
<td>$t_{\text{op}}$</td>
<td>$t_{\text{h}}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 15</td>
<td>85.5 5.33</td>
<td>85.50 0.08</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 15</td>
<td>89.5 4.37</td>
<td>91.00 0.08</td>
<td>1.67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 15</td>
<td>57.5 4.67</td>
<td>57.50 0.12</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 21</td>
<td>95.0 32.73</td>
<td>95.45 0.30</td>
<td>0.47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 21</td>
<td>116.0 31.03</td>
<td>116.03 0.28</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 28</td>
<td>121.5 132.08</td>
<td>124.47 0.53</td>
<td>2.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 28</td>
<td>136.5 184.07</td>
<td>138.91 0.47</td>
<td>1.77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 45</td>
<td>162.0 2233.17</td>
<td>165.49 2.08</td>
<td>2.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 45</td>
<td>174.0 2958.28</td>
<td>177.52 2.28</td>
<td>2.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 190</td>
<td>367.0 NA</td>
<td>397.05 51.15</td>
<td>3.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25 300</td>
<td>455.5 NA</td>
<td>471.09 226.18</td>
<td>3.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30 435</td>
<td>552.0 NA</td>
<td>559.07 748.31</td>
<td>1.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35 595</td>
<td>654.5 NA</td>
<td>678.38 1917.81</td>
<td>3.64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40 780</td>
<td>744.5 NA</td>
<td>747.25 4796.36</td>
<td>0.37</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>