The Use of Enumerative Techniques in Topological Optimization of Computer Networks Subject to Fault Tolerance and Reliability

Mostafa Abd-El-Barr and Ahmer Zakir

***Abstract*— Topological optimization of computer networks is concerned with the design of a network by selecting a subset of the available set of links such that the fault tolerance and reliability aspects are maximized while a cost constraint is met. A number of enumeration-based techniques were proposed to solve this problem. They are based on enumerating all the possible paths (for $Terminal$ reliability) and all the spanning trees (for $Network$ reliability). Existing enumeration-based techniques for solving this network optimization problem ignore the fault-tolerance aspect in their solution. Fault tolerance is an important network design aspect. A fault tolerant network is able to function even in the presence of some faults in the network.**

In this paper, we propose one algorithm for optimizing the terminal reliability and another for optimizing the network reliability while improving the fault tolerance aspects of the designed networks. Experimental results obtained from a set of randomly generated networks using the proposed algorithms are presented and compared to those obtained using the existing techniques [1], [2]. It is shown that improving the fault tolerance of a network can be achieved while optimizing its reliability however at the expense of a reasonable increase in the overall cost of the network while remaining within a maximum pre-specified cost constraint.

***Index Terms*— Fault tolerance, Reliability, Enumerative techniques, Spanning tree, Node degree.**

I. Introduction

COMPUTER networks have grown in popularity at a tremendous rate during the last decade. The advent of low-cost computing devices has led to an explosive growth in computer networks and all indications are for a continued healthy growth during the foreseeable future. One of the major advantages of computer networks over centralized systems is their ability to function even in the presence of some faults in some parts of a network.

The quality of a network is judged by its reliability and the reliability of a network depends upon the reliability of its devices (nodes), reliability of the links and the network topology. A topological design involves the determination of the links that should be established for an effective communication among the network nodes. This set of links is selected from a set of pre-specified possible links. Usually, the network topologies are fixed due to geographical or physical constraints such as in hospitals, business centers, and universities. In this situation, the problem is to choose a set of links for a given set of nodes to either maximize reliability given a cost constraint or to minimize cost given a minimum network reliability constraint [3]. If N denotes the number of nodes, the (maximum) number of links in a fully connected network is given by $N(N-1)/2$.

Some work has been done for optimizing the reliability of the network such that a cost constraint is met ([1], [4], and [2]). These techniques are based on enumeration of all the possible paths (terminal reliability) or the spanning trees (network reliability) in the network, and then they try to optimize the reliability of the network.

The authors are with the Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran - 31261, Saudi Arabia. E-mail: mostafa, azakir@ccse.kfupm.edu.sa.

II. Background Material

In this section, we provide some background material.

A computer network is modeled as a graph in which vertices (or nodes) correspond to the computers in the network and the edges correspond to the links connecting these computers. Figure 1 shows the simple case of a network consisting of four nodes and five links. Every link has a cost and reliability assigned to it, which is shown in the parentheses in Figure 1.

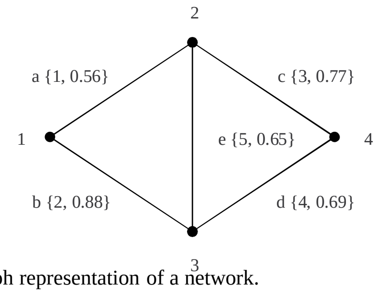

Fig. 1. Graph representation of a network.

- Definition 1: A subgraph that is a tree with no cycles and that spans (reaches out to) all vertices of the original graph is called a $Spanning\ Tree$. $\diamondsuit$

For example, links ace form a spanning tree in Figure 1.

- Definition 2: Among all the spanning trees of a weighted and connected graph, the one (possibly more) with the least total weight is called a $Minimum\ Spanning\ Tree\ (MST)$. $\diamondsuit$
- Definition 3: The distance $d(T_1, T_2)$ between any two spanning trees T_1 and T_2 is defined as:

$$d(T_1, T_2) = |T_1 - T_2| = |T_2 - T_1| \qquad (1)$$

Thus $d(T_1, T_2)$ is equal to the number of links which are present in $T_1(T_2)$ and not in $T_2(T_1)$. $\diamondsuit$

For example, Figure 1 has two spanning trees ace and bde. So the distance between these two spanning trees is 2, as two links are uncommon between them.

In designing a network, Fault Tolerance and Reliability are considered the objectives to achieve, while the Cost is considered as the constraint.

- Definition 4: $Cost$ can include material costs of the cabling, installation costs such as trenching or boring, land or right of way costs, and connection or terminal costs inherent with the cabling. $\diamondsuit$

In most of the papers, each link is assigned a weight which is used as the cost of the link.

- Definition 5: Link $Reliability$ is defined as the ability of the link to perform its function for a period of time. This reliability has a range from 0 (never operational) to 1 (perfectly reliable). $\diamondsuit$

It is assumed (with good justification) that reliability comes at a cost. The tradeoff between cost and reliability is not linear. An increase in reliability causes a greater than equivalent increase in cost [5]. Other simplifying assumptions made are that nodes are perfectly reliable and do not fail, and that links have two possible states - good or failed. Links fail independently and repair is not considered.

The reliability of a network can be seen from two different view-points [5], [2]. These are:

- Definition 6: *Terminal Reliability* is defined as the probability that a given pair of nodes in a network is connected. $\diamondsuit$
- Definition 7: *Network Reliability* is defined as the probability that all the nodes in a network are connected. $\diamondsuit$

Network reliability is concerned with the ability of each and every network node to be able to communicate with all the other nodes.

- Definition 8: A network is said to be *Fault Tolerant* if in the presence of some fault(s), data from a source to a destination can still be routed through some alternate path(s). $\diamondsuit$

For terminal reliability, we consider a network to be fault tolerant if there exists two or more totally disjoint paths between the given source-destination pair. In this case, the measure of fault tolerance is given as

$$FT = 1 - \left[\frac{\#\ of\ common\ links\ between\ paths}{Total\ \#\ of\ links\ present\ in\ the\ network}\right] \quad (2)$$

Based on this fault tolerance measure, a 1-fault tolerant network is one which retains a single established path between the source-destination pair in the presence of a fault and it can tolerate the failure of one path, as it has another path through which the destination could be reached.

While considering network reliability, we are concerned with spanning trees. As there can be no two totally disjoint spanning trees in any network, the measure for fault tolerance is given as:

$$FT = \frac{\#\ of\ nodes\ with\ node\ degree\ \geq 2}{Total\ \#\ of\ nodes\ present\ in\ the\ network} \quad (3)$$

It might happen that while the fault tolerance of a network increases, the reliability goes down a bit. A simple reason for such a scenario is that if we have two totally disjoint paths, meaning that the network is 1-fault tolerant, but the reliability of the links used in these two paths is very low, the overall reliability of the network goes down.

A. Assumptions

While considering the topological optimization of networks, we make use of the following assumptions:

- The location of the stations and the possible links are known.
- Reliability and cost of each link is known.
- Each link is bi-directional.
- Each link is either working (ON state) or failed (OFF state).
- The nodes are perfectly reliable.

B. Reliability Calculation

1) Terminal Reliability: The reliability for establishment of the initial path between a source-destination pair is computed as simple product of reliability of the links on the path. If in addition to an established path, another path is established, then the new value of reliability is calculated as follows. We apply some reductions to the network so that the reliability can be calculated much easily.

1. If there are two links $e = (i, j)$ and $f = (j, k)$ having reliabilities p_e and p_f, connecting the nodes i, j, and k respectively in series, then these two links can be replaced by a single link having reliability $p_e p_f$.

2. Two links $e = (i, k)$ and $f = (i, k)$ joining the same two nodes of the network are called parallel links. A $parallel\ reduction$ replaces two parallel links by a single link having reliability $1 - [(1 - p_e)(1 - p_f)]$.

3. Sometimes, it is not possible to reduce all graphs totally with the series-parallel reduction method, and we use the Bayes' theorem to deal with this problem. It is applied if and only if no reduction on the graph is possible. So, for a graph G, the reliability can be given as

$$R(G) = [p_e.R(G)_{e\ functions} + (1 - p_e).R(G)_{e\ fails}] \quad (4)$$

Where, $R(G)_{e\ functions}$ is the reliability of the network when link e is working, and $R(G)_{e\ fails}$ is the reliability of the network when link e has failed.

2) Network reliability: Here, reliability evaluation is done using a method proposed by Aggarwal in [6]. In this method, first all the spanning trees are enumerated using the Cartesian products of $(n - 1)$ vertex cutsets C_i whose elements are the links connected to any of the $(n - 1)$ nodes of graph G. So,

$$C = C_1 \times C_2 \times ... \times C_{n-1}$$

where C is a set of spanning trees of G with $(n - 1)$ links. The method is as follows:

1. After all the spanning trees are enumerated, a spanning tree T_0 amongst T_i's is selected and the remaining T_i's are arranged in ascending order of their distance from T_0. System success S is defined as the event of having atleast one spanning tree with all its links operative.

$S = T_0 \cup T_1 \cup ... \cup T_{n-1}$

2. Define F_i for each T_i such that:

$F_0 = T_0$

$F_i = T_0 \cup T_1 \cup ... \cup T_{i-1} \,|_{Each\ literal\ of\ T_i \rightarrow 1,\ for\ 1 \leq i \leq (n-1)}$

The literals of T_i are assigned a value 1, which is substituted in any predecessor term in which they occur.

3. Use exclusive-operator $\oplus$ to get

$$S(disjoint) = T_0 \bigcup_{i=1}^{N-1} T_i \oplus F_i \quad (5)$$

As all the terms in Equation 5 are mutually exclusive, the network reliability expression is obtained from Equation 5 by changing X_i to p_i and X_i' to q_i, so

$$R = S(disjoint)_{|X_i(X_i') \rightarrow p_i(q_i)} \quad (6)$$

C. Notation

In this section, we present the notation used in the paper.

1) Terminal Reliability Algorithms:

G	an undirected graph.
N	set of given nodes.
$NPATHS$	number of paths.
L	number of links.
P	path array, where $P(i,j) = 1$, if link j is present in path i. $P(i,j) = 0$: $i = 1, 2, ...NPATHS, j = 1, 2, ...L$.
$c(j)$	cost of link j.
p_j	probability of success of link j.
q_j	probability of failure of link j.
P_c	path cost matrix, where $P_c(i,j) = c(j)$, the cost of link j, if it exists in the path i = 0, otherwise.
P_r	path reliability matrix where, $P_r(i,j) = p_j$, the reliability of link j, if it exists in path i = 1, otherwise.
$Cost_{max}$	maximum permissible cost for the network.
$Ratio_{Disjoint}$	disjoint ratio matrix where,

$$Ratio_{Disjoint}(i) = 1 - \left[\frac{\#\ of\ common\ links\ between\ paths}{Total\ \#\ of\ links\ present\ in\ the\ network}\right].$$

$SYSCOS$	present cost of the designed system.
$SYSREL$	present reliability of the designed system.
C	column vector showing the costs of all paths where $C(i) = \sum_{j=1}^{L} P_c(i,j)$.
R	column vector giving the reliabilities of all paths where $R(i) = \prod_{j=1}^{L} P_r(i,j)$.
D	column vector with entries as the ratio of $R(i)$ and $C(i)$ $\forall\ i$, where $D(i) = \frac{R(i)}{C(i)}$.
$\triangle D(i)$	a column vector, where $\triangle D(i) = \frac{\triangle R(i)}{\triangle C(i)}$.
$\triangle R(i)$	increment in reliability of the network after adding path i.
$\triangle C(i)$	increment in cost of the network after adding path i.

2) Network Reliability Algorithms:

ST	number of spanning trees.
S_c	spanning tree cost matrix, where $S_c(k,j) = C(j)$, the cost of link j, if this link exists in the spanning tree k; = 0 otherwise. $k = 1, 2, ..., ST$.
S_r	spanning tree reliability matrix where $S_r(i,j) = p_j$; the reliability of link j, if the link exists in the spanning tree k; = 1 otherwise.
C	column vector showing the costs of all spanning trees where $C(k) = \sum_{j=1}^{L} S_c(k,j)$.
R	column vector giving the reliabilities of all spanning trees where $R(k) = \prod_{j=1}^{L} S_r(k,j)$
D	column vector with entries as the ratio of $R(k)$ and $C(k)$ for all values of k where $D(k) = \frac{R(k)}{C(k)}$.
$\triangle D(k)$	column vector, where $\triangle D(k) = \frac{\triangle R(k)}{\triangle C(k)}$.
$\triangle R(k)$	increment in reliability of the network after adding spanning tree k.
$\triangle C(k)$	increment in cost of the network after adding spanning tree k.
$Distance$	column vector, where $Distance(k)$ = Distance between the initial spanning tree and spanning tree k.
O	number of spanning trees that have been added to the network.

III. Existing Techniques and Related Work

A basic consideration in the design of a computer network is the reliable communication between some nodes, within a maximum permissible cost. The latter in turn depends upon the topological layout of the links, their costs and their reliabilities. In [1], [4], and [2], the authors have proposed three different enumerative based techniques for finding out the optimal network topology. Aggarwal and Chopra *et al.*, [4] and [1] deal with the terminal reliability while [2] deals with the network reliability. Moreover, some work has also been done on solving this problem through iterative techniques, such as Tabu Search ([7], [8]), Simulated Annealing ([9], [10]) and Genetic Algorithm (algorithms [11], [12], [13], [14], [15], [16], [17], [18], and [19]). Previous iterative based techniques are presented in another paper [20] by the authors of this paper.

Now, we present the main ideas in previous enumerative-based techniques.

A. Chopra's Terminal Reliability Technique [1]

In this approach, Chopra *et al.* proposed a technique that improves over Aggarwal's technique [4]. They do not select any links which cannot provide any additional paths between the source and the destination. The basic difference between these two techniques is that in this technique, we select a path at a time, rather than trying to add a link at a time to the already placed network [4], after the initial path is selected.

The Algorithm

Step 1: Determine all the s-t paths, assuming all possible links in position;

Step 2: Generate the path-cost matrix, P_c, and path reliability matrix, P_r;

Step 3: Generate the matrix A;

Step 4: Generate the matrix X;

Step 5: Generate the matrix D;

Step 6: Choose k such that $D(k) \geq D(i)\ \forall\ i$. Determine $A(k)$ and $X(k)$.

Step 7: Now the balance cost available is $[Cost_{max} - A(k)]$.

If $[Cost_{max} - A(k)]$ is < 0, let $D(k) = 0$ and repeat this step to find another value of k.

If $[Cost_{max} - A(k)]$ is 0, this kth path is the optimum solution, STOP.

If $[Cost_{max} - A(k)]$ is > 0, go to the next step.

Step 8: Remove the links already used from further consideration and remove any paths whose cost exceeds the balance cost available. If all the paths are removed, STOP; otherwise go to the next step.

Step 9: Generate the matrix $\triangle D(i)$.

Step 10: Choose k such that $\triangle D(k) \geq \triangle D(i) \; \forall \; i$ under consideration. Augment the network with links in path k and go back to step 7.

Example: Consider the network of Figure 2(a) as an example with the following specifications. The source s is node 5, while the destination t is node 4.

Link	a	b	c	d	e	f	g	h	i	j	k
Cost	3.30	3.70	1.35	1.25	2.55	7.95	3.0	2.0	6.0	3.0	9.15
Reliability	0.84	0.76	0.90	0.89	0.94	0.73	0.76	0.92	0.49	0.90	0.78

The total cost allowed is $Cost_{max}$ = 15 units.

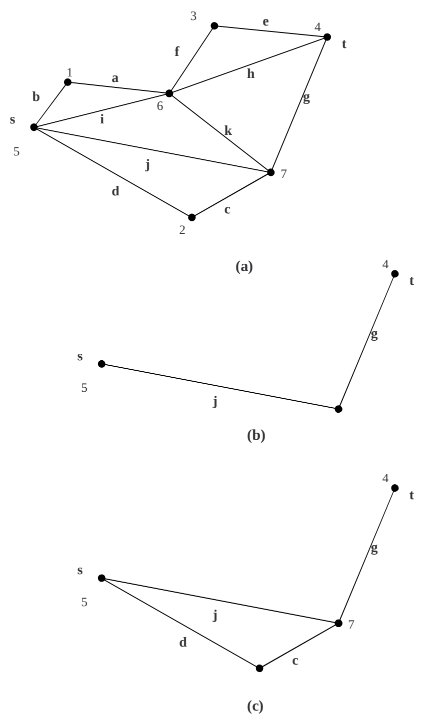

Fig. 2. Example of Chopra's terminal reliability technique.

The paths are *abef, cdg, abh, efi, hi, gj, cdefk, abgk, cdhk, gik, efjk,* and *hjk*. Now, we select the path *gj* as it has the highest reliability to cost ratio, see Figure 2(b). After placing this initial path, the network has a reliability of 0.6840. Now, we subtract the cost of the links *g* and *j*, from the cost of all the paths and ignore those paths which exceed the available cost. The reliability to cost ratio of these paths are again calculated, and the path with the highest ratio is selected. Path *cdg* is added to the initial network as it has the highest R/C ratio. The reliability of this network is 0.7449 with a cost of 8.6. Although we still have some cost available to us, but there is no path which can be added within that cost, so we stop here and finally we get the network of Figure 2(c).

B. Aggarwal's Network Reliability Technique

In this technique, the authors proposed a method for designing a computer network to maximize the *Network Reliability*. Here, the main idea is to enumerate spanning trees of the possible network topology.

The Algorithm

Step 1: Determine all the spanning trees by considering all possible links in position;

Step 2: Generate the matrix S_c;

Step 3: Generate the matrix S_r;

Step 4: Generate the matrix A;

Step 5: Generate the matrix X;

Step 6: Generate the matrix D;

Step 7: Choose k such that $D(k) \geq D(i) \; \forall \; i = 1, 2, ..., ST$. Determine $A(k)$ and $X(k)$;

Step 8: Now the balance cost available is $[Cost_{max} - A(k)]$;

If $[Cost_{max} - A(k)]$ is < 0, let $D(k) = 0$ and repeat this step to find another value of k;

If $[Cost_{max} - A(k)] = 0$, this is the optimal solution; STOP.

If $[Cost_{max} - A(k)]$ is > 0, then go to the next step;

Step 9: Remove the links already used from the spanning trees to be considered and remove all such links whose cost is greater than the balance cost available. If all the links are removed, STOP; otherwise go to the next step;

Step 10: Select k such that $\triangle D(k) \geq \triangle D(i) \; \forall \; i$ under consideration in the above step. Augment the network with link k and go back to step 8. If, however, $\triangle D(i)$ = 0 $\forall \; i$, add the first link in the network, and go back to step 8.

Example: Consider the network shown in Figure 3(a), with the following specifications:

Link	a	b	c	d	e	f	g	h
Cost	2.0	3.7	2.7	2.5	4.0	3.0	3.2	3.5
Reliability	0.9	0.6	0.8	0.5	0.9	0.7	0.7	0.8

The total cost allowed is $Cost_{max}$ = 16 units.

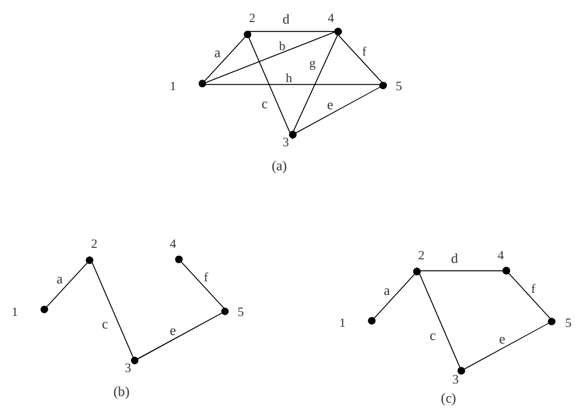

Fig. 3. Example of Aggarwal's network reliability technique.

First of all, we determine all the possible spanning trees. For this network, a total of 44 spanning trees were enumerated.

Now, we select the spanning tree *acef* as it has the highest reliability to cost ratio and so we get network as shown in Figure 3(b). After the addition of the initial spanning tree to the network, a link is added to the network at a time and the reliability to cost ratio is determined. The final network is given in Figure 3(c). Therefore, the cost of the system is 15.2 units and its reliability is 0.6250.

IV. Proposed Algorithms

As much as we have seen so far, there has been no attempt at adding the aspect of fault tolerance to a network, while we are designing it. In this section, we will present the proposed algorithms to which we have added the aspect of fault tolerance. The benefit of adding fault tolerance to any network is that if there is a failure of some link, we can still route through some alternate path or spanning tree.

For the Terminal Reliability, the idea is that after choosing the first path, we try to find *totally* disjoint path, instead of adding a path, without looking at what will be the effect of this path on the fault tolerance of the network. We start by adding the path which is totally disjoint to the already selected one, and then we continue to add lesser disjoint paths to the network, while not exceeding the cost constraint.

The same idea applies to the Network Reliability, except that here we look for as much disjoint spanning tree as possible.

A. Proposed Terminal Reliability Algorithm

We assume that the same notation is used and assumptions that were made earlier in Section II.

The Algorithm

Step 1: Determine all the source-destination paths, assuming all possible links in position;

Step 2: Generate the path-cost matrix, P_c, and path reliability matrix, P_r;

Step 3: Generate the matrix C;

Step 4: Generate the matrix R;

Step 5: Generate the matrix D;

Step 6: Choose k such that $D(k) \geq D(i) \; \forall \; i$. Determine $C(k)$ and $R(k)$;

Step 7: Compute balance cost available as $[Cost_{max} - C(k)]$;
 If $[Cost_{max} - C(k)] < 0$, let $D(k) = 0$, go to Step 6;
 If $[Cost_{max} - C(k)]$ is 0, this kth path is the optimum solution; STOP.
 If $[Cost_{max} - C(k)]$ is > 0, go to the next step;

Step 8: Remove the links already used from further consideration and remove any paths whose cost exceeds the balance cost available. If all the paths are removed, STOP; otherwise go to the next step;

Step 9: Generate matrix $\triangle D(i)$;

Step 10: Generate the matrix $Ratio_{Disjoint}$. Choose the path which has maximum value of $Ratio_{Disjoint}$. If two or more paths have the same $Ratio_{Disjoint}$, select the path which has the maximum $\triangle D(i) \; \forall \; i$ under consideration. Augment the network with links in this path and go back to step 7.

Example: Consider the network shown in Figure 4(a). The following specifications are provided for this network.

Link	a	b	c	d	e	f	g	h	i	j	k
Cost	3.30	3.70	1.35	1.25	2.55	7.95	3.0	2.0	6.0	3.0	9.15
Reliability	0.84	0.76	0.90	0.89	0.94	0.73	0.76	0.92	0.49	0.90	0.78

The total cost allowed is $Cost_{max}$ = 15 units. We would like to add fault-tolerance to this network as this aspect was overlooked previously by Chopra *et al.*

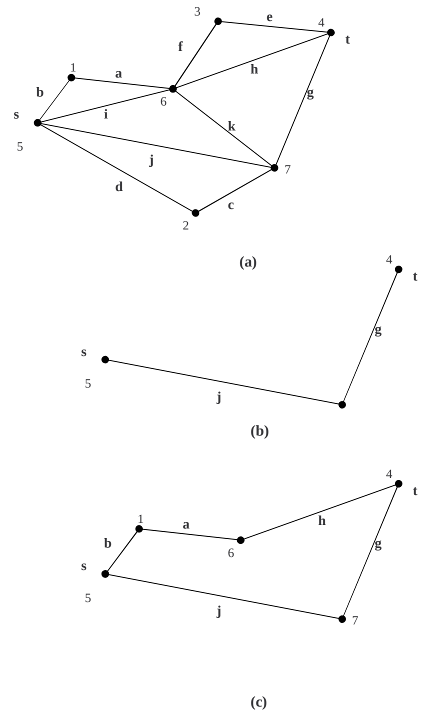

Fig. 4. Example of improved version of enumerative technique for terminal reliability.

There are 12 different paths that can be established between the source-destination pair. These are *abef, cdg, abh, efi, hi, gj, cdefk, abgk, cdhk, gik, efjk,* and *hjk*. We select the path *gj* as it has the highest reliability to cost ratio, see Figure 4(b). After placing this initial path, the terminal reliability of this network is 0.6840. After placing the initial path, the paths that can still be added to the network are: hi, cdg, and abh, as other paths' cost exceeds the balance cost.

Path	hi	cdg	abh
# of common links	0	1	0
Fault Tolerance	1.0	0.75	1.0

Now, we try to find a path which is totally disjoint from *gj*, and we select the path *abh* as it is totally disjoint from *gj*. Although the path hi is also totally disjoint from gj but the path abh yields better $\frac{\triangle R}{\triangle C}$ ratio. The final network is shown in Figure 4(c). The terminal reliability of this network is 0.8696, with a cost of 15. The benefit that we obtained by adopting this approach is that now we have 2 totally disjoint paths, which means that in the presence of some fault in a path, the other one can still be used for communication.

Observation: If we compare the final network to the network that we obtained by applying the original terminal reliability enumerative technique, we can clearly note that now the network is 1-fault tolerant which means that now we can route the traffic through two totally disjoint paths, and the failure of one path does not result in loss of communication between the source and the destination. The reliability has also increased from 0.7449 to 0.8696 at the expense of a higher cost.

B. Proposed Network Reliability Algorithm

The proposed enumerative technique for network reliability is given as follows.

The Algorithm

Step 1: Determine all the spanning trees by considering all possible links in position by the method.

Step 2: Generate S_c;

Step 3: Generate S_r;

Step 4: Generate the matrix C;

Step 5: Generate the matrix R;

Step 6: Generate the matrix D;

Step 7: Choose k such that $D(k) \geq D(i) \, \forall \, i = 1, 2, ..., ST$.

Step 8: Compute balance cost as $[Cost_{max} - C(k)]$;

If $[Cost_{max} - C(k)]$ is < 0, let $D(k) = 0$, go to Step 7;

If $[Cost_{max} - C(k)] = 0$, this is the optimal solution; STOP.

If $[Cost_{max} - C(k)]$ is > 0, go to the next step;

Step 9: Remove the links already used from the spanning trees to be considered and remove all such spanning trees whose addition is not possible. If all the spanning trees are removed, STOP; otherwise go to the next step;

Step 10: Generate the matrix $Distance$.

Step 11: Choose k such that $Distance(k) > Distance(l) \, \forall$ $l = 1, 2, ..., ST - O$. If two or more spanning trees are equally distant, select the spanning tree that makes the node degree of the nodes 2 the most.

Step 12: Augment the network with links in spanning tree k and go back to Step 7.

Example: Consider the network as shown in Figure 5(a) with the following specifications:

Link	a	b	c	d	e	f	g	h
Cost	2.0	3.7	2.7	2.5	4.0	3.0	3.2	3.5
Reliability	0.9	0.6	0.8	0.5	0.9	0.7	0.7	0.8

The total cost allowed is $Cost_{max}$ = 16 units.

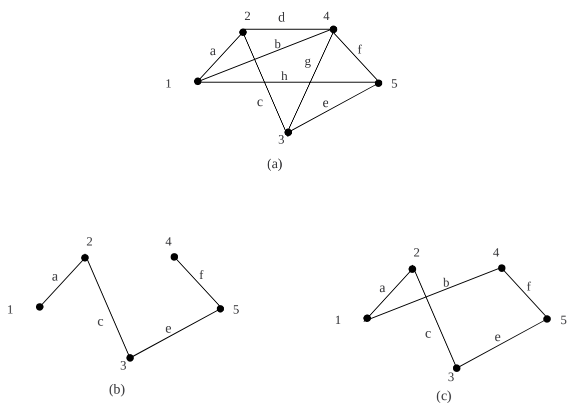

Fig. 5. Example of improved version of enumerative technique for network reliability.

We determine all the possible spanning trees, and then we select *acef* as it yields the maximum reliability to cost ratio. Now, the cost of this network is 11.7 and the reliability is 0.4536, and the network is as can be seen in Figure 5(b). Now, we try to add another spanning tree which has the highest distance from *acef* and which also does not exceed the given cost. Based on this criteria, we add *abce* as our second spanning tree and now the cost of this network is 15.4 and the network reliability of the system = 0.6685. The resultant network is shown in Figure 5(c). As there can be no subnets to add, the algorithm stops.

Observation: We have tried to maximize the fault tolerance of our network and succeeded in doing so as can be seen in Figure 5(c), as every node is now connected to atleast 2 different links, which makes the network fault-tolerant.

V. Results and Comparison

In this section, we compare the results obtained from our proposed enumerative techniques with those obtained by using the previous techniques reported in [1], [2].

A. Terminal Reliability Algorithms

We have incorporated the aspect of Fault Tolerance into our techniques, which was missing from the previous techniques [1], [2]. The results obtained from these techniques are shown in Table I.

As can be seen from the table, in most of the cases, the reliability obtained from our technique is better than that obtained from Aggarwal's method. Whereas, we were able to achieve 1-fault tolerance in almost all the cases, except for one case, because there was no other totally disjoint path available to us which could be selected.

But as could be expected, this fault tolerance comes at the expense of a greater cost, as compared to the Aggarwal's method. This seems reasonable enough because when we try to add totally disjoint path(s) to the network for making it fault-tolerant, we are adding new links to the network, which adds to the cost of the network. It can also be seen that the runtimes for both algorithms are almost equal.

B. Network Reliability Algorithms

In the network reliability algorithms, we add fault tolerance by selecting a spanning tree which is as much disjoint as possible, from the already placed spanning tree(s). The results for the previous and our proposed techniques are listed in Table II.

Here, it is observed that the fault tolerance resulting from using our technique always is equal or greater than that obtained by using the Aggarwal's method. And as seen for the terminal reliability technique, it is noted that increasing the fault tolerance of a network is synonymous to adding to cost of the network. The runtime required by the proposed algorithm is lesser than that of Aggarwal's and the reason for that is that we add a spanning tree after placing the initial spanning tree while Aggarwal adds a link at a time to the network.

TABLE I

COMPARISON BETWEEN TERMINAL RELIABILITY ALGORITHMS

Network			**Chopra's Algorithm**				**Proposed Algorithm**			
N	L	Cost_{max}	Rel	Cost	FT	Time	Rel	Cost	FT	Time
5	7	18.0	0.6734	14.0	0.75	0.33	0.7501	18.0	1.0	0.33
6	8	19.0	0.5909	14.0	0.66	0.60	0.7519	19.0	0.857	0.55
7	11	15.0	0.7449	8.6	0.75	4.51	0.8696	15.0	1.0	4.36
8	12	25.0	0.9190	23.3	0.80	9.45	0.8940	24.2	1.0	8.68
9	14	31.1	0.7011	29.8	0.80	52.89	0.7521	30.4	1.0	53.24
10	15	37.2	0.8470	33.5	0.60	73.16	0.7965	36.5	1.0	72.83
11	17	27.0	0.8441	23.8	0.75	315.0	0.8341	25.2	1.0	312.64
12	18	22.2	0.6886	20.6	0.80	668.61	0.7390	21.9	1.0	669.23
13	20	19.0	0.8421	16.3	0.75	3137.86	0.8601	18.4	1.0	3130.50
14	21	27.9	0.7272	24.1	0.80	6203.54	0.7935	27.6	1.0	6175.52
15	23	34.0	0.7374	31.6	0.60	10547.59	0.7950	34.0	1.0	10519.64

TABLE II

COMPARISON BETWEEN NETWORK RELIABILITY ALGORITHMS

Network			**Aggarwal's Algorithm**				**Proposed Algorithm**			
N	L	Cost_{max}	Rel	Cost	FT	Time	Rel	Cost	FT	Time
5	8	16.0	0.6250	15.2	0.800	0.44	0.6685	15.4	1.0	0.35
6	8	21.0	0.5599	19.0	0.833	0.35	0.6184	21.0	0.833	0.26
6	12	30.0	0.7483	29.0	0.833	2.40	0.7720	30.0	1.0	1.86
7	15	65.0	0.7014	63.4	0.714	81.97	0.7224	64.4	0.857	78.95
8	16	88.0	0.8108	85.7	0.75	105.96	0.8458	87.8	1.0	98.39
9	18	58.5	0.4836	56.6	0.888	2503.69	0.5108	58.3	0.888	2489.92
10	20	69.4	0.5477	68.5	0.800	7784.62	0.6011	69.2	1.0	7751.74
11	21	76.2	0.6741	73.5	0.818	19820.12	0.7111	75.9	1.0	19523.85

VI. CONCLUSIONS

In this work, the problem of topological optimization of computer networks subject to fault tolerance and reliability constraints is addressed. Two new enumerative techniques, one for the terminal and the other for network reliability, have been proposed and compared with the previous techniques. The results of the proposed techniques are encouraging and we are able to incorporate the issue of fault tolerance in the design process.

ACKNOWLEDGMENTS

The authors would like to acknowledge the continued support provided by the King Fahd University of Petroleum and Minerals, Saudi Arabia.

REFERENCES

[1] Y. C. Chopra, B. S. Sohni, R. K.Tiwari, and K. K. Aggarwal, "Network Topology for Maximizing the Terminal Reliability in a Computer Communication Network," *Microelectronics Reliability*, vol. 5, no. 5, pp. 911–913, 1984.

[2] K. Aggarwal, Y. Chopra, and J. S. Bajwa, "Topological Layout of Links for Optimizing the Overall Reliability in a Computer Communication System," *Microelectronics and Reliability*, vol. 22, no. 3, pp. 347–351, 1982.

[3] B. Dengiz, F. Altiparmak and A. E. Smith, "Local Search Genetic Algorithm for Optimal Design of Reliable Networks," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 3, pp. 179–188, September 1997.

[4] K. Aggarwal, Y. Chopra, and J. S.Bajwa, "Topological layout of links for optimizing the S-T reliability in a Computer Communication System," *Microelectronics and Reliability*, vol. 22, no. 3, pp. 341–345, 1982.

[5] A. E. Smith and D. L. Deeter, "A Chapter in Telecommunications Optimization: Heuristic and Adaptive Methods," .

[6] K. Aggarwal and Suraish Rai, "Reliability Evaluation in Computer Communication Networks," *Microelectronics and Reliability*, pp. 32–35, 1981.

[7] S. Pierre and A. Elgibaoui, "Improving Communication Network Topologies using Tabu Search," *22nd Annual Conference on Local Computer Networks*, pp. 44–53, November 1997.

[8] S. Pierre and A. Elgibaoui, "A Tabu Search Approach for Designing Computer Network Topologies with Unreliable Components," *IEEE Transactions on Reliability*, vol. 46, no. 3, pp. 350–359, September 1997.

[9] E. Costamagna, A. Fanni, and G. Giacinto, "A Simulated Annealing Algorithm for the Optimization of Communication Networks," *International Symposium on Signals, Systems, and Electronics, Proceedings of VLSI Systems*, pp. 405–408, October 1995.

[10] Mir M. Atiqullah and S. S. Rao, "Reliability Omptimization of Communication Networks using Simulated Annealing," *Microelectronics Reliability*, vol. 33, no. 9, pp. 1303–1319, November 1993.

[11] D. L. Deeter and A. E. Smith, "Heuristic Optimization of Network Design Considering All-Terminal Reliability," *Proceedings of the Reliability and Maintainability Symposium*, pp. 194–199, 1997.

[12] Anup Kumar, R. M. Pathak, and Y. P.Gupta, "Genetic Algorithm based Reliability Optimization for Computer Network Expansion," *IEEE Transactions on Reliability*, vol. 44, no. 1, pp. 63–72, March 1995.

[13] Jong Ryul Kim and M. Gen, "Genetic Algorithm for solving Bicriteria Network Topology Design Problem," *IEEE Proceedings of the 1999 Congress on Evolutionary Computation*, vol. 3, pp. 2272–2279, July 1999.

[14] B. Dengiz, F. Altiparmak and A. E. Smith, "Efficient Optimization of All-Terminal Reliable Networks Using an Evolutionary Approach," *IEEE Transactions on Reliability*, vol. 46, no. 1, pp. 18–26, 1997.

[15] B. Ombuki, M. Nakamura, Z. Nakao, and K. Onaga, "Evolutionary Computation for Topological Optimization of 3-connected Computer Networks," *IEEE Proceedings of Systems, Man, and Cybernetics*, pp. I–659 – I–664, 1999.

[16] F. Altiparmak and B. Dengiz, "Reliability Optimization of Computer Communication Networks using Genetic Algorithms," *IEEE Conference on Systems, Man, and Cybernetics*, vol. 5, pp. 4676–4681, October 1998.

[17] Runhe Huang and Jianhua Ma, "A Genetic Algorithm for Optimal 3-Connected Telecommunication Network Designs ," *Proceedings of the 1997 International Symposium on Parallel Architectures, Algorithms and Networks*, pp. 344–350, December 1995.

[18] J. R. Kim and M. Gen, "Genetic Algorithm for solving Bi-Criteria Network Topology Design Problem," *Proceedings of the 1999 Congress on Evolutionary Computation*, vol. 3, pp. 2272–2279, July 1999.

[19] Anup Kumar, Sanjay P. Ahuja, Chitra Sundaram, and Y. P. Gupta, "A Reliability Enhancement Approach for Computer Networks and Distributed Applications," *Twelfth Annual International Phoenix Conference on Computers and Communications*, pp. 181–187, 1993.

[20] M. Abd-El-Barr, Ahmer Zakir, Sadiq M. Sait and A. S. Almulhem, "The Use of Iterative Techniques for Topological Optimization of Computer Networks Subject to Fault Tolerance and Reliability," submitted in *IEEE Conference on Computer Communications, INFOCOM*, April 2003.