Reliability based Topological Optimization of Computer Networks - Part II: Iterative Techniques

M. Abd-El-Barr, S. M. Sait, and A. S. Al Mulhem   Ahmer Zakir

Computer Engineering Department
King Fahd University
of Petroleum & Minerals
Dhahran-31261, Saudi Arabia
mostafa, sadiq, almulhem@ccse.kfupm.edu.sa

Information Technology Center
King Fahd University
of Petroleum & Minerals
Dhahran-31261, Saudi Arabia
azakir@ccse.kfupm.edu.sa

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 enumerative and iterative based techniques were proposed to solve this problem. In this paper (Part II), we present and compare the different proposed iterative techniques for optimizing different aspects (reliability, fault tolerance, and cost) of the designed networks.

1 Introduction

One major requirement of computer networks is their ability to function even in the presence of some faults in the network. Reliable communication between some nodes within a maximum permissible cost is a basic consideration in the design of a computer network. The cost of a network depends in part on the topological layout of the links, their costs and their reliabilities. The quality of a designed network can be judged by its reliability. The reliability of a network depends upon the reliability of its nodes, reliability of the links used and the network topology. A topological design involves the determination of the sub-set of links that should be established for an effective communication among the network nodes. This sub-set of links is selected from a pre-specified set of links.

A number of iterative-based techniques for optimizing the reliability of a network has been reported in the literature. These include Tabu Search ([1], [2]), Simulated Annealing ([3], [4]) and Genetic Algorithms ([5], [6], [7], [8], [9], [10], [11], [12], and [13]). A comparison of TS, GA, and SA algorithms for topological optimization was done by Abd-El-Barr and Zakir in [14].

Iterative techniques are reviewed in Section 2. It should be noted that the background material required for this paper is provided in Part I.

2 Iterative Techniques

Kumar *et al.* [6] proposed a reliability enhancement approach for computer networks and they applied genetic algorithm (GA) to solve this problem. They have a given budget and they try to improve the reliability within this budget. One of the major advantages of their approach is that exactly the same model is used for both the terminal and network reliability. Their GA approach to this problem has two significant limitations. First, they require that all network designs considered throughout the search be feasible. While this is relatively easy to achieve using a cost constraint and a reliability objective, it might not be that easy when using a cost objective and a reliability constraint. The second limitation is their encoding, which is the difficulty in maintaining the agreement of the links present and absent after crossover and mutation.

Altiparmak *et al.* [10] used a genetic algorithm as a meta-heuristic technique for obtaining optimal or near optimal solutions to the link and node design problem. They have used integers to encode their solution to represent different types of links and nodes. A complete chromosome is divided into two fields, and the length of the chromosome is equal to the number of links plus the number of nodes. They have used three different types of links and nodes. A '1' represents most costly and reliable link or node, '2' represents second most costly and reliable link or node, and '3' represents the least reliable and costly link or node.

The objective function is the reliability of system minus a penalty function for networks that do not meet the maximum cost constraint. The objective of this penalty is to lead the algorithm to near-optimal feasible solutions, and it can be given as

$$Z(x) = R(x) - \sqrt{\frac{C(x) - Cost_{max}}{Cost_{max}}}$$

where, $Z(x)$ is the objective function, $R(x)$ is the reliability of the system and $C(x)$ is the cost of the system.

Dengiz *et al.* [8] presented a heuristic search algorithm to solve the all-terminal network design problem when considering cost and reliability. They try to minimize the cost with a reliability constraint in mind. They have used GA, but customized it appreciably for all-terminal design problem. Initial population was a set of connected networks which 2-connected with preference for solutions having high reliability. Three different reliability estimations were used to tradeoff accuracy with computation overhead. First of all, a connectivity check is done on all new networks and then networks which pass this test are subjected to 2-connectivity test. The objective function is the sum of all the links in the network, plus a penalty function for networks, which fail to meet the minimum reliability constraint, and it can be given as

$$Z(x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{i,j}.x_{i,j} + \delta(C_{max}(R(x) - R_0))^2$$

where, $c_{i,j}$ denotes the cost of the link between nodes (i, j), C_{max} is the maximum value of $c_{i,j}$, and R_0 is the network reliability requirement.

Ombuki *et al.* [9] developed a GA based technique along with a greedy heuristic to solve approximately the 3-connected network design problem. In the greedy heuristic, links are assigned to computer sites starting from a randomly selected root node and the remaining links are assigned using first assign least cost links criterion. Once we have a network with a minimum of node degree of 3, network evaluation is performed so as to avoid a situation where all links of a subset of nodes form closed groups which can result in network failure if two links fail. Their GA approach incorporates both random and least cost assignment approach with a cost function including a penalty function. They use an integer solution representation and a uniform order crossover is used for generating the new population. The objective function can be given as

$$Z(x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{i,j}.\alpha_{i,j} + T_{max}\beta$$

where, $\alpha_{i,j}\ \epsilon(0,\ 1)$ is a decision variable, β is a large positive integer and T_{max} is the number of pairs of links in the network resulting from chromosome x that makes the network disconnected if any of the pair fails. They used an arc assignment scheme which is suitable with the employed chromosome representation and their approach can be applied to both the 2-connected and 3-connected networks. Experimental study showed that the GA consistently outperformed the greedy heuristic for the problem instances that were considered.

Huang *et al.* [11] used a GA for 3-connected telecommunication network designs. Their approach uses a new method to solution representation, in which the routing, diameter, and survivability constraints can be encoded. A two-point crossover with the operation of swapping duplicated nodes ensures solutions generated through genetic evolution are all feasible so that both the checking of constraints and the repair mechanism can be avoided. The fitness of a solution is chosen as a function of the inverse of its cost and scaled into the range of [0, 1].

$$Z(x) = \frac{C_{max} - C_k}{C_{max} - C_{min}}$$

where, C_k is the cost of the current solution, while C_{max} and C_{min} are the maximum and minimum cost of solutions in a population, respectively.

Kim *et al.* [12] presented a GA based approach for solving the bicriteria network design problems to optimality, or near-optimality. Because of the feature that only a spanning tree topology can be used as active network configurations, they employed an encoding method that employs Prüfer number and cluster string in order to represent a chromosome. The Prüfer number is capable of uniquely representing all possible spanning trees and it can also contain the information pertaining to the node degree of any node. An M/M/1 model is used to describe a single cluster behavior and the bicriteria network topology design is formulated as the following non-linear 0-1 programming model:

$$minimize\ \frac{1}{\Gamma}[\sum_{i}^{n} \frac{F_i(X)}{C_i - F_i(X)} + \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij}.f_{ij}(X)]$$

$$minimize\ \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{1ij}.x_{1ij} + \sum_{i=1}^{n} \sum_{j=1}^{m} w_{2ij}.x_{2ij}$$

$$subject\ to\ R(x) > R_{min}$$

where Γ is the total offered traffic, β_{ij} is the delay/bit between center i and j, C_i is the traffic capacity of center i, w_{1ij} is the cost of the link between centers i and j, w_{2ij} is the cost of the link between center i and user j, and $R(x)$ is the network reliability.

Deeter and Smith [5] proposed a heuristic approach to design of networks when considering network reliability formulated as minimizing cost given a reliability constraint. The approach is demonstrated on multiple test problems with varying degree of constraint. Each element of the chromosome represents a possible link and the value of each of these elements tells what type of connection each link has with the connected nodes. Selection for the next generation is done using a rank based quadratic procedure. They have used uniform crossover by selecting two parent solutions and then randomly taking a component from one parent to form the corresponding component of the child.

Abd-El-Barr and Zakir [14] have proposed a GA-based technique in which they have incorporated the aspect of fault tolerance along with reliability into the design process, whereas cost acts as a constraint. They used the weighted-sum approach to aggregate the two objectives of fault tolerance and reliability in a single measure. Based on this weighted sum technique, the problem takes the following form:

$$Overall\ value = (w_f * Fault\ Tolerance) + (w_r * Reliability)$$

The weights used for simulations are 0.6 for the fault tolerance and 0.4 for reliability. The latter is calculated by using Jan's method [15]. They also developed an expression for measuring the fault tolerance of a network.

$$Fault\ Tolerance = \frac{\#\ of\ Nodes\ with\ node\ degree\ \geq\ 2}{Total\ \#\ of\ Nodes\ in\ the\ network}$$

A node degree ≥ 2 is desired in order to guarantee that in the event of one link failing, the node can still communicate with rest of the network using the other available link.

A chromosome is a binary string of 0's and 1's. Its length is equal to the number of links in the network. A 1 in some location in this string means that the link is present, whereas a 0 represents that the link is not present. The initial solutions are generated randomly. They have used two-point crossover with a crossover probability of 0.996 and mutation is performed with a probability of 0.04.

Example:

For this example, we have a population size N_p of 4, and the number of offsprings N_o is 2. This means that every solution has to take part in crossover, which is shown in Figure 1.

After applying the crossover, we apply mutation on the offsprings as shown in Figure 2. Now, we have 6 solutions in total and since N_p is 4, we select 4 out of these 6 solutions using the *Selection* mechanism discussed above. The final result for this iteration is as shown in Figure 3. These solutions become the starting point for the next iteration and this process continues till 10,000 iterations are completed.

Atiqullah and Rao [4] presented a simulated annealing algorithm which selects the optimal set of links

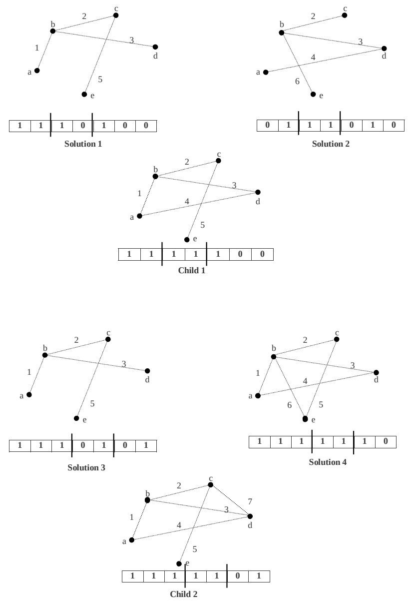

Figure 1: Example for GA.

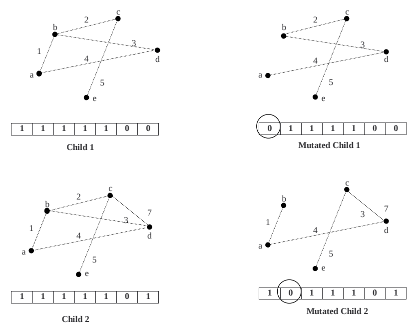

Figure 2: Example for GA (Contd.).

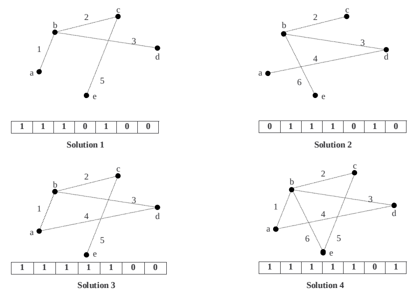

Figure 3: Example for GA (Contd.).

that maximized the network reliability, subject to a cost constraint. Their algorithm employs a variation of the simulated annealing approach coupled with a hierarchical strategy to achieve the global optimum. The major modification is the elimination of the exponential acceptance criteria, which means no randomly selected design is accepted unless there is a definite improvement over the previous design. There are two stages in their algorithm. In the first stage, the cost and reliability of are evaluated. The second stage systematically allows a random perturbation of the current design and passes the new design through a series of logical acceptance tests. Although their algorithm used the concept of simulated annealing, but no cooling schedule was utilized since they do not accept any worse design probabilistically.

Abd-El-Barr and Zakir [14] proposed an approach based on simulated annealing for solving the topological optimization problem. An initial random solution is generated and then a perturbation is applied on the current solution in order to generate a new solution. This is done by creating a neighbor of the current solution by either adding or removing a link from the current solution. However, if a link is removed from the network, the solution is checked for validity, i.e., if the network is still connected or not.

Empirically, the total allowed time for the annealing process was set at 4000. Temperature is initialized to a value T_0 at the beginning, and is slowly reduced; the parameter α is used to achieve this cooling. The cooling rate α was set at 0.988. The amount of time spent in annealing at a given temperature is gradually increased as temperature is lowered. This is done using the parameter $\beta \geq 1$, therefore the value for β was set as 1.005.

Example:

SA is a much simpler algorithm as compared to GA and TS. Figure 4 (a) shows the current solution, while Figure 4 (b) shows the solution after link 6 has been removed from the current solution. If the resulting solution is better than the previous one, the move is accepted, otherwise it has to satisfy the $Metropolis$ criteria to get accepted. In this example, the topology in Figure 4 (b) becomes the current solution for next iteration. This procedure is repeated until we reach the $Maxtime$.

Costamagna [3] also applied the Simulated Annealing algorithm to our problem. A feasible configuration is represented with a binary string of n elements: a '1' in a certain position of the string means that the multiplexer has been activated in the corresponding node; '0' means that the node does not contain a multiplexer center. The initial temperature t_0 is chosen in the following manner: let χ be the acceptance ratio, i.e. the ratio between the number of accepted solutions and the number of generated solutions, the value of t_0 is doubled until χ becomes higher than a fixed value of χ_0. The final value of temperature is strictly linked to stopping criterion.

Pierre ([1], [2]) presented a tabu search approach for designing computer network topologies using unreliable components. Some moves or local transformations called *Perturbations* are applied to a starting topology in order to reduce its total link cost and/or to improve its average packet delay. These perturbations deal with addition, removal, and substitution of links. A fixed tabu list size of 7 was used with a neighborhood size of 6.

Abd-El-Barr and Zakir [14] also proposed a technique based on the Tabu Search technique to solve topological optimization problem. Here, the solution is encoded in the same manner as for GA implementation. In each iteration, a number of neighbors of the current solution are generated by making perturbations, with the probability of adding a link higher than removing it. They have used a neighborhood size of 8 for experiments.

The characteristic of the move stored in tabu list is the index of link. The decision regarding the size of tabu list affects time and space requirements of TS. Empirically, we have used a tabu list size of $\lfloor \sqrt{L/2} \rfloor$. The aspiration criteria used is the following: if the best neighbor solution of the current iteration is better than the global best solution, then tabu list restrictions are overridden and the solution is accepted. Also the global best solution and the aspiration criteria is updated. The algorithm was run for 10,000 iterations.

Example:

First of all, we have to make 8 neighbors of the solution, as the neighborhood size is set at 8. Each neighbor is generated by making a single perturbation to the current solution, as shown in Figure 5. Since L = 7 in this example, the TL size is 1.

After we have generated the neighbors, the best neighbor is selected and checked if the move corresponding to that neighbor is in the tabu list. If the move does not exist in tabu list, we accept that neighbor as our current solution for the next iteration and update the tabu list and the aspiration criteria. If the

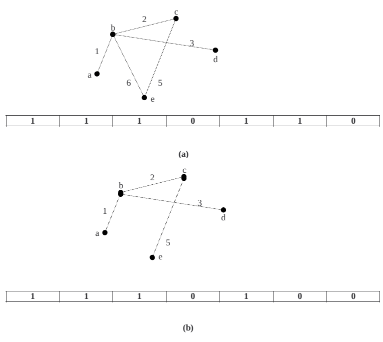

Figure 4: Example for SA.

move happens to be in the tabu list and the aspiration criteria is also not satisfied, the current solution does not change and we go to the next iteration and the same process is repeated. Final solution for this iteration of TS is as shown in Figure 6.

3 Conclusions

In this paper, a review of the work which has been done to solve the problem of topological optimization of computer networks is presented. Different techniques are discussed and compared which aim at finding an optimal solution (topology) where reliability and fault tolerance are taken as objectives and cost is the constraint.

Acknowledgments

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

References

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

[2] S. Pierre and A. Elgibaoui, "A Tabu Search Approach for Designing Computer Network Topologies with Unreliable Components," *IEEE Transactions on Reliability*, vol. 46, pp. 350–359, September 1997.

[3] 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.

[4] M. M. Atiqullah and S. S. Rao, "Reliability Omptimization of Communication Networks using Simulated Annealing," *Microelectronics Reliability*, vol. 33, pp. 1303–1319, November 1993.

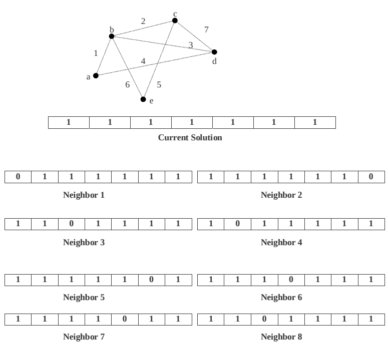

Figure 5: Example for TS.

[5] 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.

[6] Anup Kumar, R. M. Pathak, and Y. P.Gupta, "Genetic Algorithm based Reliability Optimization for Computer Network Expansion," *IEEE Transactions on Reliability*, vol. 44, pp. 63–72, March 1995.

[7] J. R. 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.

[8] 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.

[9] 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.

[10] 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.

[11] R. Huang and J. 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.

[12] 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.

[13] 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.

[14] M. Abd-El-Barr, Ahmer Zakir, Sadiq M. Sait and A. S. Almulhem, "Iterative Heuristics in Topological Optimization of Computer Networks Subject to Fault Tolerance and Reliability," submitted to *The Eighth IEEE Symposium on Computers and Communications (ISCC'2003)*, July 2003.

[15] Rong-Hang Jan, "Design of Reliable Networks," *Computers and Operations Research*, vol. 20, no. 1, pp. 25–34, 1993.

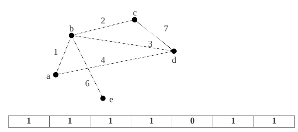

Figure 6: Example iteration of TS (Contd.).