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

Mostafa Abd-El-Barr*, Ahmer Zakir**, Sadiq M. Sait*, and Abdulaziz Almulhem

Abstract— **Topological optimization of computer networks is concerned with the selection of a subset of the available links such that the reliability and fault-tolerance aspects are maximized while meeting a cost constraint. In this case, the problem is stated as optimizing the reliability and fault-tolerance of a network subject to a maximum cost constraint. Existing iterative-based techniques consider the simple single-objective version of the problem by considering reliability as the only objective. We consider fault-tolerance to be an important network design aspect.**

We consider the use of three iterative techniques, namely *Tabu Search*, *Simulated Anealing*, and *Genetic Algorithms*, in solving the multi-objective topological optimization network design problem. Experimental results for a set of 10 randomly generated networks using the three iterative techniques are presented and compared. 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.

Keywords: Topological Optimization, Fault Tolerance, Reliability, Genetic Algorithm, Tabu Search, Simulated Annealing, Spanning tree.

I. Introduction

A topological design involves selection of a 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 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 to connect a given set of nodes to either maximize reliability given a cost constraint or to minimize cost given a minimum network reliability constraint [1].

The topological optimization design problem when considering only reliability has also been studied in the literature using alternative methods of search and optimization. Previous iterative approaches include Tabu Search [2], Genetic Algorithms [3], [4], [5], [6], [7], [8], [9], [10], and Simulated Annealing [11], [12].

In this paper, we obtain a solution to the multiobjective network design problem by optimizing both the reliability and fault-tolerance of a network subject to a maximum cost constraint. It should be noted that the background material required for this paper is provided in Part I.

In the following, we present our implementation of the three algorithms for the topological optimization of computer networks problem.

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

II. Proposed Iterative Techniques

In this section, we present the basic heuristic and implementation details of the three iterative algorithms (GA, TS, and SA) and their parameters selection.

A. Genetic Algorithm

GA is an elegant search technique that emulates the process of natural evolution as a means of progressing towards the optimal solution. The structure of GA for topological optimization problem is given in Figure 1.

1) Chromosomal Representative: It is a binary string of 0's and 1's whose length equals 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 solution is generated randomly.

2) Fitness Evaluation: The weighted sum technique to aggregate the two objectives of fault tolerance and reliability into a single measure is used. Based on this weighted sum technique, the problem takes the following form.

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

The weights used for simulations are 0.6 for fault tolerance and 0.4 for reliability. The latter is calculated by using Jan's method [13].

3) Parent Choice: Parent selection is done using *roulette wheel* in our implementation.

4) Crossover and Mutation: We have applied two-point crossover, and after applying the crossover, the child solution is checked for validity. Mutation is applied with a certain probability to the offsprings generated as a result of the crossover. A mutation rate of 0.04 was used in this work.

5) Stopping Criterion: The experiments were carried out for different number of total generations. It is observed that after certain number of generations, GA tends to converge and there is no further significant improvement in the solution quality. As a result, it is decided to use a fixed number of 10,000 iterations as stopping criterion for GA.

0-7803-7978-0/03/$17.00 ©2003 IEEE

Algorithm *Genetic Algorithm*;
(*N_p is the Population size*)
(*N_g is the number of Generations*)
(*N_o is the number of Offsprings*)
(*P_c is the Crossover probability*)
(*P_μ is the Mutation probability*)
Begin
Construct Initial Population of size N_p;
For (j = 1 to N_p)
Evaluate Fitness (Population[j])
End For
For (i = 1 to N_g)
For (j = 1 to N_o)
Select Parents for Crossover and Mutation
(x, y) ← Parents;
Perform Crossover with probability P_c;
Offspring[j] ← Crossover(x, y);
Perform Mutation with probability P_μ;
/*by adding or removing a link*/;
Evaluate Fitness (Offspring[j])
End For
Population ← Select (Population, Offspring, N_p);
End For
Return highest scoring configuration in the population;
End(*of Genetic Algorithm*)

Fig. 1. Genetic Algorithm for Topological Optimization.

Algorithm *Tabu Search*;
(*Ω is the set of feasible solutions*)
(*S is the Current solution*)
(*S^* is the Best solution*)
(*$Cost_{Overall}$ is the Objective function*)
(*$N(S)$ is the neighborhood of S*)
(*V^* is the sample of N(S)*)
(*TL is the Tabu list*)
(*AC is the Aspiration Criteria*)
Begin
Start with an initial feasible solution(topology);
Initialize TL and AC
For Fixed number of Iterations **Do**
Generate Neighbor solutions V^*;
/*by adding or removing a link*/
Find best $S^* \in N(S)$;
If move S to S^* is not in TL **Then**
Accept move and update S^*;
Update TL and AC;
Else
If $Cost_{Overall}$ > AC **Then**
Update TL and AC;
End If
End If
End For
End(*of Tabu Search*)

Fig. 2. Tabu Search for Topological Optimization.

B. Tabu Search (TS)

Tabu search is an iterative heuristic that has been applied for solving a range of combinatorial optimization problems in different fields [14]. The structure of TS for topological optimization problem is given in Figure 2.

1) Initialization: The solution encoding and initialization steps are similar to those described for GA implementation. The only difference is that a single random initial solution is created, in contrast to GA, where multiple initial solutions are generated.

2) Neighborhood Generation: 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. In this work, a neighborhood size of 8 was used.

3) Tabu List and Aspiration Criterion: The purpose of Tabu list is to avoid revisiting a point (solution) in the search space. In the present implementation, the characteristic of the move stored in tabu list is the index of link. We have used a tabu list size of $\sqrt{L/2}$.

The aspiration criterion 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.

4) Stopping Criterion: TS was run for different number of total iterations, and depending on experimental observations, it is decided to run the algorithm for a fixed number of 10,000 iterations.

C. Simulated Annealing (SA)

Simulated Annealing is another optimization technique that falls in the category of general adaptive heuristics [14].

The Simulated annealing algorithm for topological optimization problem is shown in Fig. 3, whereas the Metropolis procedure is shown in Fig. 4.

1) Initial Solution: An initial solution is chosen randomly which must correspond to a valid connected network topology.

2) Perturb Function: After the generation of the initial solution, the Metropolis procedure is called to apply the *Perturb* function 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.

III. Results and Comparison

In this section, we compare the results obtained using GA, TS, and SA.

The networks with the specifications shown in Table I were used for simulations in our experiments. The $Cost_{max}$ column shows the maximum cost constraint, which has to be observed while designing a network.

A. Tabu Search and Simulated Annealing

Here, the performance of TS is compared with that of SA. The results shown are the best case results obtained by best possible tuning of various algorithmic parameters of TS and

TABLE II

TS Vs SA COMPARISON

Network		Tabu Search						Simulated Annealing					
No.	$Cost_{max}$	Rel	FT	Cost	Overall	T_{best}	T_{tot}	Rel	FT	Cost	Overall	T_{best}	T_{tot}
1	130.0	0.803	1.0	128.4	**0.9213**	2	5	0.783	1.0	128.9	**0.9131**	5	11
2	140.0	0.820	1.0	139.9	**0.9280**	3	12	0.810	1.0	138.6	**0.9239**	5	19
3	190.0	0.799	1.0	185.2	**0.9196**	2	16	0.794	1.0	188.1	**0.9175**	4	22
4	280.0	0.862	1.0	278.7	**0.9448**	8	24	0.846	0.975	279.3	**0.9235**	21	38
5	330.0	0.892	1.0	329.9	**0.9569**	9	26	0.884	1.0	329.7	**0.9538**	10	50
6	460.0	0.906	1.0	458.8	**0.9622**	8	35	0.904	0.983	449.8	**0.9517**	13	63
7	520.0	0.757	1.0	515.3	**0.9026**	14	43	0.719	1.0	515.6	**0.8875**	22	81
8	700.0	0.904	1.0	699.1	**0.9618**	12	60	0.817	1.0	697.5	**0.9269**	16	97
9	680.0	0.600	1.0	673.0	**0.8399**	28	84	0.552	1.0	675.6	**0.8208**	31	130
10	850.0	0.747	1.0	848.1	**0.9028**	48	94	0.716	1.0	848.2	**0.8863**	51	175

Algorithm $Sim.\ Annealing(S_0, T_0, \alpha, \beta, M, Maxtime)$;
($*S_0$ is the initial solution*)
(*BestS is the best solution*)
($*T_0$ is the initial temperature*)
($*\alpha$ is the cooling rate*)
($*\beta$ is a constant*)
($*Maxtime$ is the total allowed time *)
($*M$ is the time until the next parameter update*)
Begin
$T = T_0$;
$CurS = S_0$;
$BestS = CurS$; /*$BestS$ is the best solution*/
$CurCost = Cost(CurS)$;
$BestCost = Cost(BestS)$;
$Time = 0$;
Repeat
Call $Metropolis(CurS, CurCost, BestS, BestCost, T, M)$;
$Time = Time + M$;
$T = \alpha T$;
$M = \beta M$;
Until $(Time \geq Maxtime)$;
Return$(BestS)$
End.($*of\ Simulated\ Annealing$*)

Fig. 3. Simulated Annealing for Topological Optimization.

Algorithm $Metropolis(CurS, CurCost, BestS, BestCost, T, M)$;
Begin
Repeat
$NewS = Perturb(CurS)$;
/*by adding or removing a link*/
$NewCost = Cost(NewS)$;
$Gain = (NewCost - CurCost)$;
If $Gain > 0$ **Then**
$CurS = NewS$;
If $NewCost > BestCost$ **Then**
$BestS = NewS$;
$BestCost = NewCost$;
EndIf
Else
If $(RANDOM < e^{-Gain/T})$ **Then**
$CurS = NewS$;
$CurCost = NewCost$;
EndIf
EndIf
$M = M - 1$;
Until $M = 0$
End ($*of\ Metropolis$*)

Fig. 4. The Metropolis procedure.

TABLE I

NETWORKS USED FOR SIMULATIONS

Network (Test Cases)			
No.	N	L	$Cost_{max}$
1	20	30	130.0
2	25	38	140.0
3	30	60	190.0
4	40	80	280.0
5	50	100	330.0
6	60	120	460.0
7	70	140	520.0
8	80	160	700.0
9	90	180	680.0
10	100	200	850.0

SA. The results for the best solutions generated by TS and SA are listed in Table II.

As can be seen from the *Overall* column in Table II, the TS algorithm performed better than SA in every case, with better or equal fault tolerance and reliability values. Moreover, the time taken to achieve the best solution using TS is shorter than that taken by SA to achieve the best solution.

Figures 5, 6, 7 show the *Overall* results for network 7 in Table II using TS, SA, and GA, respectively.

B. Simulated Annealing and Genetic Algorithm

In this section, we compare the results obtained using SA with those obtained using GA. The results obtained are as listed in Table III.

From the results, it is clear that SA performed better than GA for all the circuits in terms of quality of best solution as well as run time.

IV. CONCLUDING REMARKS

In this work, the problem of topological optimization of computer networks considering fault tolerance and reliability as objectives and cost as the constraint is addressed. Iterative algorithms provide powerful solutions for solving hard problems with large search space, in comparison to enumerative techniques, which are not practical for large size networks.

Three iterative algorithms, namely Tabu Search, Simulated Annealing, and Genetic Algorithm are presented and compared

TABLE III

SA Vs GA Comparison

Network		Simulated Annealing						Genetic Algorithm					
No.	$Cost_{max}$	Rel	FT	Cost	Overall	T_{best}	T_{tot}	Rel	FT	Cost	Overall	T_{best}	T_{tot}
1	130.0	0.783	1.0	128.9	**0.9131**	5	11	0.780	0.950	126.8	**0.8820**	7	21
2	140.0	0.810	1.0	138.6	**0.9239**	5	19	0.802	0.960	136.3	**0.9004**	10	29
3	190.0	0.794	1.0	188.1	**0.9175**	4	22	0.790	0.967	185.5	**0.8960**	14	40
4	280.0	0.846	0.975	279.3	**0.9235**	21	38	0.842	0.975	268.1	**0.9216**	25	59
5	330.0	0.884	1.0	329.7	**0.9538**	10	50	0.887	0.960	326.3	**0.9287**	35	83
6	460.0	0.904	0.983	449.8	**0.9517**	13	63	0.902	0.983	457.8	**0.9508**	51	108
7	520.0	0.719	1.0	515.6	**0.8875**	22	81	0.697	0.986	517.0	**0.8702**	81	150
8	700.0	0.817	1.0	697.5	**0.9269**	16	97	0.808	0.988	696.8	**0.9159**	88	181
9	680.0	0.552	1.0	675.6	**0.8208**	31	130	0.547	0.989	677.2	**0.8120**	57	249
10	850.0	0.716	1.0	848.2	**0.8863**	51	175	0.711	0.980	846.5	**0.8722**	116	362

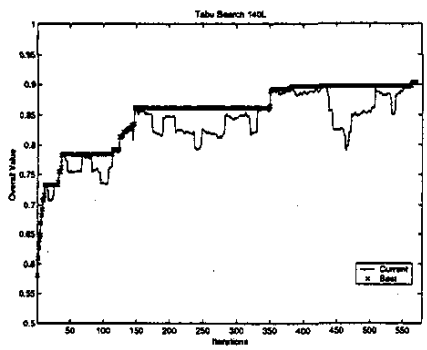

Fig. 5. Tabu Search – Overall (for network 7).

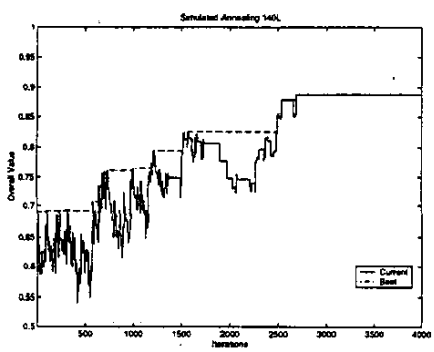

Fig. 6. Simulated Annealing – Overall (for network 7).

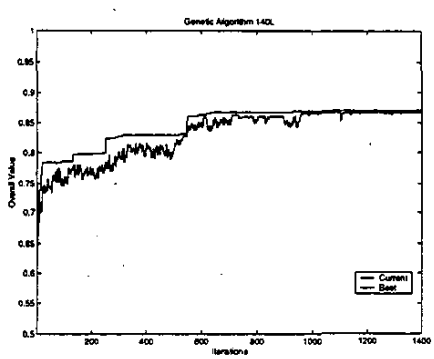

Fig. 7. Genetic Algorithm – Overall.

for solving this hard problem. It is shown that *Tabu Search* outperforms *Simulated Annealing* and *Genetic Algorithms* in terms of the quality of solution and runtime. The results of the above techniques are promising and show that these are well engineered for the topological optimization problem.

ACKNOWLEDGMENTS

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

REFERENCES

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] B. Ombuki, M. Nakamura, Z. Nakao, and K. Onaga, "Evolutionary Computation for Topological Optimization of 3-connected Computer Networks," *IEEE*, 1999.

[8] 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.

[9] 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.

[10] 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.

[11] 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 on VLSI Systems*, pp. 405–408, October 1995.

[12] 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.

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

[14] Sadiq M. Sait and Habib Youssef, *Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems*, IEEE Computer Society Press, California, December 1999.