GENETIC ALGORITHM DESIGN OF NETWORKS CONSIDERING ALL-TERMINAL RELIABILITY

Berna Dengiz and Fulya Altiparmak
Department of Industrial Engineering
Gazi University
06570 Maltepe, Ankara Turkey

Alice E. Smith[1]
Department of Industrial Engineering
University of Pittsburgh
Pittsburgh, PA 15261 USA
email: aesmith@engrng.pitt.edu

Abstract - The use of computer communication networks has been rapidly increasing recently to 1) share expensive hardware and software resources, and 2) provide access to main systems from distant locations. The reliability and the cost of these systems are important considerations that are largely determined by placement of the nodes and the links between nodes. In this study, a genetic algorithm (GA) is presented to solve the all-terminal network design problem when considering cost and reliability. The GA is considerably enhanced over conventional implementations to improve effectiveness and efficiency. This general optimization approach is shown to be computationally efficient and highly effective on a large suite of test problems with search spaces up to 2×10^{90}.

Keywords - network design, network reliability, genetic algorithm, all-terminal reliability

1. INTRODUCTION

The design of reliable communication networks is a significant problem in the telecommunications industry. An important stage of network design is to find the best layout of components to minimize cost while meeting a performance criterion, such as transmission delay, throughput or reliability [14]. Generally, a large scale network has a multilevel, hierarchical structure consisting of a backbone network and several local access networks [2]. This paper is focused on large scale backbone communication network design where the relevant reliability metric is all-terminal network reliability (also called overall network reliability); defined as the probability that every pair of nodes can communicate with each other [5, 14]. The design approach focuses on selection of link topology given fixed and perfectly reliable nodes. The problem of optimal design of link topology can be formulated as a combinatorial problem where the selection of components either maximizes reliability given constraints, or minimizes cost given a minimum network reliability constraint. This problem is NP-hard [9], the search space growing exponentially with the number of nodes. Further compounding this growth in possible topological architectures is the computational effort required to calculate or estimate network reliability.

This problem has been studied in the literature with both enumerative based methods (usually a variation of branch-and-bound) [14] and heuristic methods [1, 3, 18]. This paper uses a genetic algorithm (GA) approach as did [7, 15, 16], but significantly customizes it to the all-terminal design problem. This customization results in an effective and efficient

1 Corresponding author.

optimization methodology. Furthermore, the approach in this paper is demonstrated on a large test suite of problems, that includes large networks with up to 300 possible links. Previous work, including those cited above, has demonstrated the heuristic and exact optimization procedures on small networks (usually less than 10 nodes), thus the important issue of scale-up is left unanswered.

Notation

N	set of nodes (terminals)
L	set of links (edges, arcs)
(i,j)	a link between nodes i and j
p	link reliability
x_{ij}	decision variable, $x_{ij} \in \{0,1\}$
$\mathbf{x}$	a link topology of $\{x_{11}, x_{12}, \ldots, x_{ij}, \ldots, x_{N,N-1}\}$
$R(\mathbf{x})$	all-terminal reliability of $\mathbf{x}$
R_o	network reliability requirement
Z	objective function
c_{ij}	cost of link (i,j)
g	generation number of GA
n	population size of GA
r_c	crossover rate of GA
r_m	mutation rate of GA

2. STATEMENT OF THE PROBLEM

The following define the problem assumptions:

1. The location of each network node is given.
2. Nodes are perfectly reliable.
3. Each c_{ij} and p are fixed and known.
4. Each link is bi-directional.
5. Only one link is possible between i and j.
6. Links are either operational or failed.
7. The failure of links are independent.
8. No repair is considered.

The optimization problem is:

$$\text{Minimize} \quad Z = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij} x_{ij} \qquad (1)$$

$$\text{Subject to :} \quad R(\mathbf{x}) \geq R_o$$

When minimizing cost subject to a minimum reliability constraint, the network reliability calculation is necessary to determine feasibility of the candidate design. If the state occurs that any pair of nodes cannot communicate because of one or more failed links, the network is considered failed. Summing this state occurrence probability over all operational states gives the all-terminal network unreliability. However, the exponential number of states makes such a computation infeasible, even for networks of moderate sizes [6]. An efficient Monte Carlo simulation technique by Yeh et al. [19] is used to estimate all-terminal reliability of a network in this paper. To reduce the computational effort required, each network must be at least a spanning tree with 2-connectivity. For some networks, an upper bound on network reliability by Jan [13] is used in lieu of Monte Carlo simulation. However, any method of calculating or estimating network reliability could be incorporated into the evolutionary optimization methodology.

3. SOLUTION ALGORITHM

Genetic algorithms were selected as the heuristic optimization vehicle because of their flexibility and robustness as demonstrated on many NP-hard problems. GA is a meta-heuristic inspired by the biological paradigm of evolution. They were pioneered by Holland [11], De Jong [8], and Goldberg [10] in the context of continuous non-linear optimization, and later extended by various authors to combinatorial problems. In GA, the search space is composed of candidate solutions to the problem, each represented by a string, termed a chromosome. Each chromosome has an objective function value, called the fitness. A set of chromosomes together with their associated fitness is called the population. This population, at a given iteration of the genetic algorithm, is referred to as a generation.

There are three main steps in the repeat loop for GA: 1) The process of selecting strings from the current generation to be parents of the next generation with preference for fitter strings. This is the **selection** process for reproduction. 2) The process of combining two selected strings to generate new children strings, which is called **crossover**. Probabilistically, components of a chromosome are perturbed while generating a child. This process is called **mutation**. Together, crossover and mutation comprise **reproduction**. 3) Computation of the **fitness** value using the objective function of each new solution. The steps in the GA approach in this research are discussed below, followed by a flowchart of the algorithm.

3.1. Coding Structure

Each string, $\mathbf{x}$, represents a candidate network design with the size of the string equal to N(N-1)/2, the number of possible links in a fully connected network. An element, x_{ij}, of **x** represents a link between N_i and N_j coded using a 0,1 alphabet. For example, a simple network whose base graph consists of 5 nodes and 10 possible links, but with only 7 links present, is shown in Figure 1.

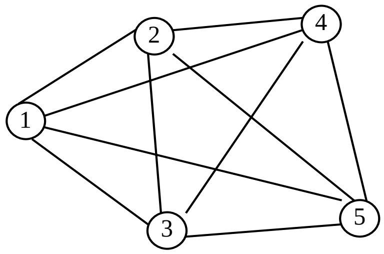

Figure 1: Typical network.

The string representation of this network can be seen below:

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} x_{12} & x_{13} & x_{14} & x_{15} & x_{23} & x_{24} & x_{25} & x_{34} & x_{35} & x_{45} \end{bmatrix}$$

3.2. Initial Population

The initial population consists of a set of connected networks which are also 2-connected but is otherwise generated in a random fashion with preference to combinations yielding high reliability. The 2-connectivity measure is used as a preliminary screening, since it is usually a property of highly reliable networks.

3.3. Genetic Algorithm Parameters

The conventional GA operators of roulette wheel selection, single point crossover and bit flip mutation were used. See Goldberg [10] for definitions of these. The choice of parameters for GA can affect performance of the algorithm. These parameters include population size (n), crossover rate (r_c) and mutation rate (r_m). For this study, it was found that good results over all network sizes were achieved with $n = 20$, $r_c = 0.95$ and $r_m = 0.05$.

3.4. Objective and Fitness Functions

The objective function is the sum of the total cost for all links in the network plus a quadratic penalty function for networks which fail to meet the minimum reliability requirement. The objective of the penalty function is to lead the optimization algorithm to near-optimal, feasible solutions. It is important to allow infeasible solutions into the population because good solutions are often the result of breeding between a feasible and an infeasible solution and the genetic algorithm reproduction procedure does not ensure feasible children, even if both parents are feasible [4], especially in highly constrained problems where the constraint is likely to be active. The fitness function considering possible infeasible solutions is given by:

$$Z(\mathbf{x}) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij}\, x_{ij} + \delta\, (c_{MAX}(R(\mathbf{x}) - R_o))^2 \quad (3)$$

$$\delta = \begin{cases} 0, & \text{if } R(\mathbf{x}) \geq R_o \\ 1, & \text{if } R(\mathbf{x}) < R_o \end{cases}$$

c_{MAX} = the maximum value of c_{ij}

In GA, the fitness, $F(\mathbf{x})$, traditionally improves with an improved objective function, creating a maximization problem. Therefore, the fitness in this paper is defined as:

$$F(\mathbf{x}) = (Z_{MAX} - Z(\mathbf{x})) \quad (4)$$

where Z_{MAX} is the largest (worst) value of equation 3 for the current population.

3.5. Network Reliability

The GA methodology in this paper uses three reliability estimations to trade-off accuracy with computational effort. An ideal strategy would only employ the computationally intensive method of Monte Carlo simulation (or exact network reliability calculation) on the optimal network design. Since GA is an iterative algorithm, this ideal cannot be attained as many candidate networks must be evaluated during the search. Therefore, screening of candidate network designs is used in the GA. First, a connectivity check for a spanning tree is made on all new network designs using the method of [12]. Then, for networks which pass this check, the 2-connectivity measure [17] is made by counting the node degrees. Finally, for networks which pass both of these preliminary checks, Jan's upper bound [13] is used to compute the upper bound of reliability of a candidate network, $R_U(\mathbf{x})$. This upper bound in used in the calculation of the objective function (eq. 3) for all networks except those which are the best found so far ($\mathbf{x}_{BEST}$). Networks which have $R_U(\mathbf{x}) \geq R_o$ and the lowest cost so far are sent to the Monte Carlo subroutine for more precise estimation of network reliability using an efficient Monte Carlo technique by Yeh et al. [19]. The simulation is done for 3000 iterations for each candidate network for all problems studied in this paper.

3.6. Termination Condition

The criterion is the total number of generations, g_{MAX}, where g_{MAX} varies according to the size of the network, N, under study.

3.7. GA Pseudo-Code

Step 1: Generate the initial population, $k = 1$ to n, randomly, discarding any solutions which fail to meet the spanning tree and the 2-connectivity requirements. Calculate the fitness of each candidate network in the population using eq. 4 and Jan's upper bound as R(**x**), except for the lowest cost network with $R_U(\mathbf{x}) \geq R_o$. For this network, $\mathbf{x}_{BEST}$, use the Monte Carlo estimation of R(**x**) in eq. 4. $g = 1$.
Step 2: Select two candidate networks from current population by the selection mechanism.
Step 3: To obtain two children candidate networks, apply reproduction to the selected networks using r_c and r_m.
Step 4: Discard any child that is not a spanning tree with 2-connectivity.
Step 5: Calculate $R_U(\mathbf{x})$ for each child and compute its objective function using eq. 3.
Step 6: If the number of new children < n-1 go to Step 2.
Step 7: Replace parents with children, retaining $\mathbf{x}_{BEST}$ from the previous generation.
Step 8: Sort the new generation in increasing order of Z with k the indices of a candidate network. $k = 1$ to n.
a) If $Z(\mathbf{x}_k) < Z(\mathbf{x}_{BEST})$, then estimate the system reliability of this network using Monte Carlo simulation, else go to Step 9.
b) $\mathbf{x}_{BEST} = \mathbf{x}_k$. Go to Step 9.
Step 9 : Calculate the fitness value, F(**x**), using eq. 4 of each network in the new population.
Step 10 : If $g = g_{MAX}$ stop, else go to Step 2 and $g = g+1$.

4. RESULTS AND CONCLUSIONS

The test problems are summarized in Table 1 and details are available from the authors. These problems are both fully connected and non-fully connected networks. N of the connected networks ranges from 5 to 25. The available links of the non-fully connected networks were randomly generated and were 1.5 times the number of nodes, N. For these networks, the encoding was reduced from N(N-1)/2 to the number of possible links. The link costs for all networks were randomly generated over [1,100] except for problems 3 through 5 which used costs over [1,150]. Each problem for the GA was run 10 times, each time with a different random number seed.
The comparison of the GA over the 10 seeds (best, mean and coefficient of variation) with the optimal solutions obtained by the method of Jan et al. [14] is given in Table 1. Jan's method cannot be practically used on the larger problems (numbers 15-17) because of the computational effort of the branch-and-bound procedure. As shown, the GA gives the optimal value for the all replications of problems 1, 2 and 3 and finds optimal for all but two of the problems for at least one run of the 10. The two with suboptimal results (12 and 13) are very close to optimal.

Table 2 lists the search space for each problem along with the proportion actually searched by the GA during a single run ($n \times g_{MAX}$). g_{MAX} ranged from 30 to 20000, depending on problem size. This proportion is an upper bound because GA's can (and often do) revisit solutions already considered earlier in the evolutionary search. It can be seen that the GA approach examines only a minute fraction of the possible solutions for the larger problem, yet still yields optimal or near-optimal final solutions.

Table 2 also makes a comparison of the efficacy of the Monte Carlo estimation of network reliability. The exact network reliability is calculated using a backtracking algorithm also used by Jan et al. [14] and compared to the estimated counterpart for the final network for those problems where the GA found optimal. It can be seen that the reliability estimation of the Monte Carlo method is unbiased and is always within 1% of the exact network reliability. Since the computation time for the Monte Carlo method is constant with network size and estimation accuracy does not degrade with an increase in L, this simulation estimation is a very effective surrogate for an exact network reliability calculation.

In this study, a stochastic search algorithm based on GAs was developed to solve network topology design with minimum cost subject to a reliability constraint. The strengths of this evolutionary approach are almost non-increasing computational effort, effective optimization and flexibility. Since GA is an iterative algorithm and improvement is typically diminishing, it may be terminated at any time and still return good results. The computational effort over the test problems studied did not vary significantly although network size increased by many orders of magnitude. The GA returned optimal or near-optimal solutions on every run regardless of problem instance, problem size or random number seed. The methodology is very flexible and alternative objectives (e.g., maximize reliability subject to a cost constraint) and alternative methods of reliability calculation (e.g., backtracking or another Monte Carlo method) could be easily substituted for those used in this research.

ACKNOWLEDGMENTS

Part of this study was funded by The Scientific and Technical Research Council of Turkey (STRCT). Alice Smith gratefully acknowledges support of the National Science Foundation CAREER grant DMI 95-02134.

REFERENCES

[1] K. K. Aggarwal, Y. C. Chopra and J. S. Bajwa, "Topological layout of links for optimising the overall reliability in a computer communication system," *Microelectronics and Reliability*, Vol. 22 (1982), 347-351.

[2] R. R. Boorstyn and H. Frank, "Large scale network topological optimization," *IEEE Transactions on Communications*, Com-25 (1977), 29-7.

[3] Y. C. Chopra, B. S. Sohi, R. K. Tiwari and K. K. Aggarwal, "Network topology for maximizing the terminal reliability in a computer communication network," *Microelectronics and Reliability*, Vol. 24 (1984), 911-913.

[4] D. W. Coit, A. E. Smith and D. M. Tate, "Adaptive penalty methods for genetic optimization of constrained combinatorial problems," *INFORMS Journal on Computing*, Vol. 8 (1996), 173-182.

[5] C. J. Colbourn, *The Combinatorics of Network Reliability*, Oxford University Press (1987).

[6] C. J. Colbourn, "Combinatorial aspects of network reliability," *Annals of Operations Research*, Vol. 33 (1991), 3-15.

[7] D. L. Deeter and A. E. Smith, "Heuristic optimization of network design considering all-terminal reliability," *Proceedings of the Reliability and Maintainability Symposium* (1997), 194-199.

[8] K. A. DeJong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, Ph.D. Thesis, University of Michigan (1975).

[9] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., San Francisco (1979).

[10] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA. (1989)

[11] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor (1975).

[12] J. E. Hopcroft and J. D. Ullman, "Set merging algorithms," *SIAM Journal of Computers*, Vol. 2 (1973), 296-303.

[13] R.-H. Jan, "Design of reliable networks," *Computers and Operations Research*, Vol. 20 (1993), 25-34.

[14] R.-H. Jan, F.-J. Hwang, and S.-T. Chen, "Topological optimization of a communication network subject to a reliability constraint," *IEEE Transactions on Reliability*, Vol. 42 (1993), 63-70.

[15] A. Kumar, R. M. Pathak, Y. P. Gupta and H. R. Parsaei, "A genetic algorithm for distributed system topology design," *Computers and Industrial Engineering*, Vol. 28 (1995), 659-670.

[16] A. Kumar, R. M. Pathak and Y. P. Gupta, "Genetic-algorithm-based reliability optimization for computer network expansion," *IEEE Transactions on Reliability*, Vol. 44 (1995), 63-72.

[17] L. G. Roberts and B. D. Wessler, "Computer network development to achieve resource sharing," *Proceedings of the Spring Joint Computer Conference of AFIPS* (1970), 543-549.

[18] A. N. Venetsanopoulos and I. Singh, "Topological optimization of communication networks subject to reliability constraints," *Problem of Control and Information Theory*, Vol. 15 (1986), 63-78.

[19] M.-S. Yeh, J.-S. Lin and W.-C. Yeh, "A new Monte Carlo method for estimating network reliability," *Proceedings of the 16th International Conference on Computers & Industrial Engineering* (1994), 723-726

BIOGRAPHICAL SKETCHES

Berna Dengiz is an Associate Professor of Industrial Engineering and Vice Dean of the Faculty of Engineering. Her research has been sponsored by The Scientific and Technical Research Council of Turkey. Dr. Dengiz is a member of Society for Computer Simulations, Operations Research Society of Turkey, Informatics Society of Turkey and Statistics Society of Turkey.

Fulya Altiparmak is a Research Assistant of Industrial Engineering. She received a B.Sc. (1987), M.Sc. (1990) and Ph.D. (1996) in Industrial Engineering from Gazi University. She is a member of the Operations Research Society of Turkey.

Alice E. Smith is Associate Professor of Industrial Engineering and Board of Visitors Faculty Fellow at the University of Pittsburgh. Her research has been funded by Lockheed Martin Corporation, ABB Daimler Benz Transportation (ADtranz), the National Institute of Standards (NIST), the Ben Franklin Technology Center of Western Pennsylvania and the National Science Foundation (NSF). She is a Senior Member of

IIE, IEEE and SWE, a member of INFORMS and ASEE, and a Registered Professional Engineer.

TABLE 1: Comparison of Results of GA over 10 Seeds with Optimal Design.

Problem	N	L	p	R_o	Optimum Cost [14]	Results of Genetic Algorithms Best	Mean	CV
FULLY CONNECTED NETWORKS								
Problem 1	5	10	0.80	0.90	255	255	255.0	0
Problem 2	5	10	0.90	0.95	201	201	201.0	0
Problem 3	7	21	0.90	0.90	720	720	720.0	0
Problem 4	7	21	0.90	0.95	845	845	857.0	0.0185
Problem 5	7	21	0.95	0.95	630	630	656.0	0.0344
Problem 6	8	28	0.90	0.90	203	203	205.4	0.0198
Problem 7	8	28	0.90	0.95	247	247	249.5	0.0183
Problem 8	8	28	0.95	0.95	179	179	180.3	0.0228
Problem 9	9	36	0.90	0.90	239	239	245.1	0.0497
Problem 10	9	36	0.90	0.95	286	286	298.2	0.0340
Problem 11	9	36	0.95	0.95	209	209	227.2	0.0839
Problem 12	10	45	0.90	0.90	154	156	169.8	0.0618
Problem 13	10	45	0.90	0.95	197	205	206.6	0.0095
Problem 14	10	45	0.95	0.95	136	136	150.4	0.0802
Problem 15*	15	105	0.90	0.95	---	317	344.6	0.0703
Problem 16*	20	190	0.95	0.95	---	926	956.0	0.0304
Problem 17*	25	300	0.95	0.90	---	1606	1651.3	0.0243
NON FULLY CONNECTED NETWORKS								
Problem 18	14	21	0.90	0.90	1063	1063	1076.1	0.0129
Problem 19	16	24	0.90	0.95	1022	1022	1032.0	0.0204
Problem 20	20	30	0.95	0.90	596	596	598.6	0.0052

TABLE 2: Search Effort of the GA and Reliability Estimation Performance.

Problem	Search Space	Solutions Searched	Fraction Searched	R_o	Actual R(x)	Estimated R(x)	Percent Difference
Problem 1	1.02 E3	6.00 E2	5.86 E-1	0.90	0.9170	0.9170	0.000
Problem 2	1.02 E3	6.00 E2	5.86 E-1	0.95	0.9579	0.9604	0.261
Problem 3	2.10 E6	1.50 E4	7.14 E-3	0.90	0.9034	0.9031	-0.033
Problem 4	2.10 E6	1.50 E4	7.14 E-3	0.95	0.9513	0.9580	0.704
Problem 5	2.10 E6	1.50 E4	7.14 E-3	0.95	0.9556	0.9569	0.136
Problem 6	2.68 E8	2.00 E4	7.46 E-5	0.90	0.9078	0.9078	0.000
Problem 7	2.68 E8	2.00 E4	7.46 E-5	0.95	0.9614	0.9628	0.001
Problem 8	2.68 E8	2.00 E4	7.46 E-5	0.95	0.9637	0.9645	0.083
Problem 9	6.87 E10	4.00 E4	5.82 E-7	0.90	0.9066	0.9069	0.033
Problem 10	6.87 E10	4.00 E4	5.82 E-7	0.95	0.9567	0.9545	-0.230
Problem 11	6.87 E10	4.00 E4	5.82 E-7	0.95	0.9669	0.9668	-0.010
Problem 12	3.52 E13	8.00 E4	2.27 E-9	0.90	0.9050	*	
Problem 13	3.52 E13	8.00 E4	2.27 E-9	0.95	0.9516	*	
Problem 14	3.52 E13	8.00 E4	2.27 E-9	0.95	0.9611	0.9591	-0.208
Problem 15*	4.06 E31	1.40 E5	3.45 E-27	0.95	@	0.9509	
Problem 16*	1.57 E57	2.00 E5	1.27 E-52	0.95	@	0.9925	
Problem 17*	2.04 E90	4.00 E5	1.96 E-85	0.90	@	0.9618	
Problem 18	2.10 E6	1.50 E4	7.14 E-3	0.90	0.9035	0.9035	0.000
Problem 19	1.68 E7	2.00 E4	1.19 E-3	0.95	0.9538	0.9550	0.126
Problem 20	1.07 E9	3.00 E4	2.80 E-5	0.90	0.9032	0.9027	-0.055

* Optimal not found by GA.

@ Network is too large to exactly calculate reliability.