Journal of Heuristics, 9: 471–487, 2003

Optimal Design of Reliable Computer Networks: A Comparison of Metaheuristics

FULYA ALTIPARMAK AND BERNA DENGIZ
Department of Industrial Engineering, Gazi University, Ankara, Turkey

ALICE E. SMITH
Department of Industrial and Systems Engineering, Auburn University, Auburn, Alabama, USA
email: aesmith@eng.auburn.edu

Submitted in August 2001 and Accepted by Steve Chiu in September 2003 after 2 revisions

Abstract

In many computer communications network design problems, such as those faced by hospitals, universities, research centers, and water distribution systems, the topology is fixed because of geographical and physical constraints or the existence of an existing system. When the topology is known, a reasonable approach to design is to select components among discrete alternatives for links and nodes to maximize reliability subject to cost. This problem is NP-hard with the added complication of a very computationally intensive objective function. This paper compares the performance of three classic metaheuristic procedures for solving large and realistic versions of the problem: hillclimbing, simulated annealing and genetic algorithms. Three alterations that use local search to seed the search or improve solutions during each iteration are also compared. It is shown that employing local search during evolution of the genetic algorithm, a memetic algorithm, yields the best network designs and does so at a reasonable computational cost. Hillclimbing performs well as a quick search for good designs, but cannot identify the most superior designs even when computational effort is equal to the metaheuristics.

Key Words: hillclimbing, simulated annealing, genetic algorithms, memetic algorithm, network reliability, network design

1. Introduction

One of the major advantages of a computer network over a centralized system is its potential for improved system reliability. The reliability of a network depends on the reliabilities of its nodes and communication links, and on network topology. Network reliability is usually characterized by either "terminal reliability", i.e. the probability of finding a path between two specified nodes, or by "overall reliability", i.e. the probability of network connectedness. Overall reliability is also termed global availability and is defined as the probability that the network is at least connected, that is, it at least forms a spanning tree. This parameter is an appropriate measure of computer communication network reliability if all users are to be provided with the possibility of being connected with each other.

The network design problem involving overall reliability is normally cast as maximizing reliability subject to a cost constraint, or minimizing cost subject to a reliability constraint.

There are two main difficulties with this problem class. First, the calculation of overall network reliability is used in either formulation, and is in itself, NP-Hard (Colbourn, 1987; Provan and Ball, 1982). Most papers in the literature have adopted surrogates to reliability, such as topological requirements, to circumvent this computational difficulty. However, these surrogates are just that, and can be too approximate or even misleading in some cases. The second difficulty in the problem class is the usual one of combinatorial problems–size grows exponentially with nodes and links and choices of components for each. In the literature, to reduce the combinatorial size of the problems, it has generally been assumed that nodes are perfectly reliable so that the decision problem is only concerned with the links, and most researchers have further assumed that only one choice of link component is available. The transformed problem then becomes where to place the links. While this formulation is more computationally tractable, it lacks fidelity with the original design problem.

In surveying the relevant literature, versions of the design problem are generally solved heuristically. Jan, Hwang, and Chen (1993) developed an algorithm using decomposition based on branch and bound to minimize link costs with a minimum network reliability constraint. Aggarwal, Chopra, and Bajwa (1982) maximized overall reliability given a cost constraint for networks with differing link reliabilities using a greedy heuristic. Venetsanopoulos and Singh (1986) used a two-step procedure for the problem of minimizing a network's cost subject to a reliability constraint. The algorithm first used a heuristic to develop an initial feasible network configuration, then a branch and bound approach was used to improve this configuration. A deterministic version of simulated annealing (SA) was used by Atiqullah and Rao (1993) to find the optimal design of small networks (five nodes or less). Pierre et al. (1995) also used SA to find optimal designs for packet switch networks where delay and capacity were considered, but reliability was not. Tabu search (TS) was used by Glover, Lee, and Ryan (1991) to choose network design when considering cost and capacity, but not reliability. Another TS approach by Beltran and Skorin-Kapov (1994) was used to design reliable networks by searching for the least cost spanning 2-tree, where the 2-tree requirement was a surrogate for reliability. Koh and Lee (1995) also used TS to find telecommunication network designs that required some nodes (special offices) have more than one link while others (regular offices) required only one link, also using this link constraint as a surrogate for network reliability. Kumar et al. (1995) developed a genetic algorithm (GA) considering diameter, average distance, and computer network reliability and applied it to four test problems of up to nine nodes. They calculated overall network reliability exactly and used a maximum network diameter (minimal number of links between any two nodes) as a constraint. The same authors used this GA to expand existing computer networks (Kumar, Pathak, and Gupta, 1995). Davis et al. (1993) approached a related problem considering link capacities and re-routing upon link failure using a customized GA. Abuali, Schoenefeld, and Wainwright (1994a, 1994b) assigned terminal nodes to concentrator sites to minimize costs while considering capacities using a GA, but reliability was not considered. Deeter and Smith (1998) presented a GA approach for the minimum cost network design problem with perfectly reliable nodes, alternative link reliabilities and an overall network reliability constraint. Dengiz, Altiparmak, and Smith (1997a, 1997b) addressed the same network design problem as in Deeter and Smith (1998). using a fairly standard GA implementation in the former paper and an efficient problem-specific GA in the latter paper.

The papers cited above deal almost exclusively with greenfield network design, that is, link locations (and, in a few cases, types) are selected (assuming nodes are not part of the decision problem). However, with many network design problems, such as those at industrial parks, hospitals, schools, laboratories and government installations, network topology is fixed because of geographical and physical constraints or existence of an existing system. When the topology is known, the problem of optimal design can be defined as the selection of types of components (links and nodes) among alternatives to maximize reliability subject to cost. This problem is not harder or easier than the greenfield design. It is identical in that network reliability must be calculated (or estimated) for each candidate design. It is different in that locations are not among the decisions, but node types are included as decision variables along with link types. This problem of design for a fixed topology was addressed by Altiparmak, Dengiz, and Smith (1998, 2000) using a GA to select link and node types for relatively small networks.

Because of the computational impracticality of using exact optimization methods for problems of realistic size, only heuristics are considered in this paper, as in the bulk of the literature cited above. Generic forms of hillclimbing (HC), SA and GA metaheuristics are developed. In addition, hybrid forms of SA and GA, using HC to improve random solutions, are also investigated. Comparisons using problems of realistic size, including an actual design problem for a major Turkish university, have been carried out considering both solution quality and computational efficiency and recommendations are made as to the best format of heuristic for this problem class. This paper not only approaches a somewhat different network design problem than is usually studied, but also makes a useful comparison among the heuristics.

Notation:

G	a probabilistic graph
N	set of given nodes
L	set of given links
m	set of link types
k	set of node types
l_j	jth link type, $j \in m$
n_i	ith node type, $i \in k$
p_{lj}	reliability of jth link type (probability that it is operational)
p_{ni}	reliability of ith node type (probability that it is operational)
c_{lj}	unit cost of jth link type
c_{ni}	cost of ith node type
d_j	distance of link j
$\mathbf{x}$	topology of a network
$\mathbf{y}$	set of decision variables for selection of link types in $\mathbf{x}$
$\mathbf{z}$	set of decision variables for selection of node types in $\mathbf{x}$
$R(\mathbf{x})$	overall reliability of $\mathbf{x}$
C_0	maximum cost constraint
$C(\mathbf{x})$	total cost of $\mathbf{x}$

Z objective function
π current solution
$f(\pi)$ objective function value of π

Note that when a mission time of the network is not specified, as in this paper, reliability (node, link and network) is identical to stationary availability.

2. Problem formulation

A communication network can be modeled by a probabilistic graph $G = (N, L, p_{ni}, p_{lj})$ in which N and L are the set of nodes and links that correspond to computer sites and communication links with reliabilities respectively p_{ni} and p_{lj}. The networks here are assumed to have bi-directional links and it is further assumed that there are no parallel (i.e., redundant) links.[1] Failures are assumed to be independent and components can only take one of the two states, operational or failed. No repair is considered.

The optimization problem is:

$$\begin{aligned} &\text{Maximize} \quad R(\mathbf{x}; \mathbf{y}, \mathbf{z}) \\ &\text{Subject to} \quad C(\mathbf{x}; \mathbf{y}, \mathbf{z}) \leq C_0 \end{aligned} \tag{1}$$

From (1), the problem is the selection of types of components (links and nodes) among their alternatives in a fixed network topology ($\mathbf{x}$) to maximize reliability subject to cost. Although network topology is fixed, the total cost and reliability of the network changes based on the selected link and node types. Compounding the difficulty of the optimization problem is the calculation of network reliability in the objective function value. At any time instant, only some links or nodes of G might be operational. A state of G is a sub-graph (N, L') where L' is the set of operational (not failed) links such that $L' \subseteq L$. The overall network reliability of state $L' \subseteq L$ considering node reliability is:

$$R(\mathbf{x}) = \left(\sum_{\Omega} \left[\prod_{i \in L'} \sum_{j=1}^{|m|} p_{l_j} y_{ij} \right] \cdot \left[\prod_{i \in (L \setminus L')} \sum_{j=1}^{|m|} \left(1 - p_{l_j}\right) y_{ij} \right] \right) \cdot \quad \prod_{i \varepsilon N} \sum_{j=1}^{|k|} p_{n_j} z_{ij} \tag{2}$$

where:

$$\sum_{j=1}^{|m|} y_{ij} = 1, \quad i = 1, 2, \ldots, |L|$$

$$\sum_{j=1}^{|k|} z_{ij} = 1, \quad i = 1, 2, \ldots, |N|$$

$$y_{ij} \in \{0, 1\} \quad \forall\, i, j$$

$$z_{ij} \in \{0, 1\} \quad \forall\, i, j$$

and where Ω is the set of all operational states. In other words, all nodes of the network must be connected through at least one path (the links form at least a spanning tree of the network). This also requires that all nodes be operating, as evident in the last term of Eq. (2).

Overall reliability is synonymous with stationary availability in the absence of a mission time.

There are two approaches to calculate overall network reliability: simulation and analytic. Analytic methods grow exponentially with network size because they depend on the enumeration of Ω. However, when the number of nodes and links are identical, that is $|N| = |L|$, the backtracking algorithm of Ball and Van Slyke (1977) and Jan (1993) to calculate overall reliability can be modified to:

$$R(\mathbf{x}) = \left(\prod_{i=1}^{|L|} \sum_{j=1}^{|m|} p_{l_j} y_{ij} + \sum_{s=1}^{|L|} \left[\prod_{\substack{i=1 \\ i \neq s}}^{|L|} \sum_{j=1}^{|m|} p_{l_j} y_{ij} \right] \sum_{j=1}^{|m|} (1 - p_{ls}) y_{sj} \right) \prod_{i=1}^{|N|} \sum_{j=1}^{|k|} p_{n_j} z_{ij} \quad (3)$$

which is computationally expedient and exact. The networks considered in this paper are of this sort, excepting the final network, a small actual design problem. It must be noted that networks with equal cardinality of nodes and links contain exactly one cycle and are not typical of data communications networks, which usually contain multiple cycles. Since an exact calculation may not practical for some larger problems due to computational effort, simulation methods could be used (see for example, Nel and Colbourn, 1990; Yeh, Lin, and Yeh, 1994) or upper or lower bounds (see for example, Ball and Provan, 1983; Brown, Colbourn, and Devitt, 1993; Nel and Colbourn, 1990) or even neural network estimation (Srivaree-ratana, Konak, and Smith, 2002). Depending on the size of the network design problem and the computational resources available, the user can decide on which method (or methods) of reliability calculation or estimation to be used.

The total cost of a network is calculated by:

$$C(\mathbf{x}) = \left[\sum_{i \in L} \sum_{j=1}^{|m|} c_{l_j} d_i y_{ij} + \sum_{i \in N} \sum_{j=1}^{|k|} c_{n_j} z_{ij} \right] \quad (4)$$

Because heuristics can benefit from consideration of both feasible and infeasible solutions during constrained search (Smith and Coit, 1997), if a design is infeasible (exceeds the cost constraint), it is penalized according to the amount exceeded:

$$\begin{aligned} Z(\mathbf{x}) &= \max((R(\mathbf{x}) - P(\mathbf{x})), 0) \\ P(\mathbf{x}) &= \begin{cases} \sqrt{\left[\dfrac{(C(\mathbf{x}) - C_0)}{C_0} \right]}, & \text{if } C(\mathbf{x}) > C_0 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (5)$$

This is a straightforward penalty that depends on the amount of constraint violation which is known to be advantageous in heuristic search (Smith and Coit, 1997).

3. Solution algorithms

Three metaheuristics, hillclimbing (HC), simulated annealing (SA) and genetic algorithms (GA), and hybrid forms of SA and GA were selected due to the large search space and the nonlinear objective function. While these methods have been applied to versions of the network design problem before, they have not been systematically studied comparing both computational effort and solution quality. Furthermore, versions of SA and GA that employ HC within the search have not been compared to the classic versions. Each metaheuristic is briefly introduced below, followed by the specific encoding and operators for this problem.

Hillclimbing. A simple approach for solving optimization problems is HC, sometimes more accurately termed "steepest descent (or ascent)" (Aarts and Lenstra, 1997; Pirlot, 1992). HC uses an iterative improvement technique and starts from an arbitrary solution $\pi \in S$, the search space. At each iteration, a new solution π' is chosen in the neighborhood of the current solution π. This implies the definition of a neighborhood structure on S: for each $\pi \in S$ a subset $V(\pi) \subseteq S$ is called the neighborhood of π. By convention, it is assumed that no solution is a neighbor of itself, i.e. $\pi \notin V(\pi), \forall \pi \in S$. While a neighborhood might be defined in almost any way for a given problem, normally neighboring solutions will be in proximity (in some manner) to the current solution. Proximity might be measured in problem space, in encoding space or even in solution space. The most common way of choosing a new solution π' in the neighborhood of π is to pick (one of) the best one(s), i.e. a solution $\pi' \in V(\pi)$ with $f(\pi') \geq f(\pi)$ (for maximization problem), $\forall \pi' \in V(\pi)$. Then π' becomes the next current solution if it is not worse than π, i.e. $f(\pi') \geq f(\pi)$. Otherwise, the search is terminated. HC provides only local optimum values and these values depend on the selection of the initial solution, π_o.

Simulated annealing. The interest in SA for optimization began with the work of Kirkpatrick, Gerlatt, and Vecchi (1983) and it has been used successfully for many problems as reviewed in Eglese (1990), Koumolos, Antony, and Jean (1994) and Aarts and Lenstra (1997). Starting from an initial solution, π_o, SA generates a new solution, π', in the neighborhood of the original solution with a specified generation method. The amount of change in the objective function value, $\Delta = f(\pi') - f(\pi)$, is calculated. If $\Delta \geq 0$, then a move to the new solution is accepted (for a maximization problem). When $\Delta < 0$, the move is accepted with a specified probability, usually denoted by $\exp^{-|\Delta|/T}$, where T is a control parameter which corresponds to temperature in the analogy of a physical annealing schedule. T is generally monotonically reduced over the search.

Genetic algorithm. GA was pioneered by Holland (1975), DeJong (1975), and Goldberg (1989) in the context of continuous non-linear optimization, and later extended to combinatorial optimization problems. Each chromosome (solution) has an objective function value, termed the fitness. A set of chromosomes together with their associated fitness is the population. This population, at a given iteration of the GA, is a generation. The key elements of the algorithm are crossover and mutation operators. Crossover combines the features of multiple parent structures to form offspring. Mutation randomly alters one or more components of a selected structure. The population evolves through subsequent generations with preference for fitter solutions.

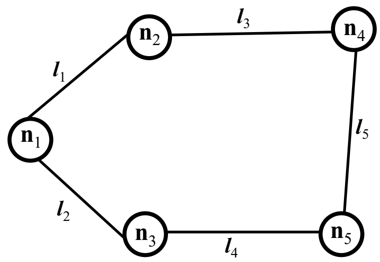

Figure 1. A network topology with five nodes and five links.

3.1. Implementation of the metaheuristics

Integers were used to encode the design problem and each integer represented a type of link or node. The complete string is divided into two fields so that the length of the string is equal to the number of links plus the number of nodes in a given network. The first $|L|$ and next $|N|$ numbers in the string represent the assigned type of links and nodes, respectively. For example; figure 1 shows a simple network whose graph consists of 5 nodes and 5 links.

Using three types of links and nodes, 1 represents the most reliable and costly link (node), 2 represents the second most reliable and costly link (node), and so on. An example string for the network given in figure 1 is:

$$\begin{array}{ccccc|ccccc} l_1 & l_2 & l_3 & l_4 & l_5 & n_1 & n_2 & n_3 & n_4 & n_5 \\ 1 & 2 & 1 & 2 & 3 & 2 & 2 & 1 & 2 & 3 \end{array}$$

Links' Field Nodes' Field

When the links' field in the example string above is examined, it can be seen that the most reliable and costly link type, represented by 1, is assigned to links l_1 and l_3. The second most reliable and costly link type, represented by 2, is assigned to links l_2 and l_4. The same representation is valid for the nodes' field. This representation has a length of $|L| + |N|$ and can easily be extended to include any number of types of links or nodes. It can also accommodate fixed link or node types (i.e., links or nodes that are not part of the decision problem).

The objective function is Eq. (5), where $R(\mathbf{x})$ is calculated by Eq. (3), and the cost is calculated using Eq. (4). Initial solutions are randomly generated. All heuristics, except HC, are terminated according to a maximum number of solutions considered that varied with problem size (200 K for 20 nodes, 300 K for 35 nodes and 400 K for 50 nodes).

3.1.1. Hillclimbing. The procedure of HC is as follows. Fifty initial solutions are generated and the best among these is taken as the starting point. The neighborhood is obtained by altering the value of an integer in the string by one, exhaustively. Therefore this neighborhood is comprised of solutions that differ by only link or node, and that difference is only by one value. The solution that has the largest increase in reliability and also satisfies the cost constraint becomes the new current solution. If no solution in the neighborhood improves reliability and meets the cost constraint, the search is terminated, that is, a local optimum is reached.

3.1.2. Simulated annealing. The initial solution in SA is obtained in same manner as HC. A move in SA is realized using following scheme: a number is randomly selected between 1 and $|L| + |N|$. This number represents the link or node that will change. Next, a new type of the link (node) is randomly selected between 1 and m (1 and k) excepting its current type. After changing the current type of the link (node) to the new type, the new solution becomes the current solution. T_o was determined using the empirical rule of Kirkpatrick, Gerlatt, and Vecchi (1983) to be 0.6 resulting in the initial fraction of accepted downhill moves of approximately 0.85. As a cooling schedule, a single iteration at each temperature, as per Lundy and Mees (1986), was used. The temperature at each iteration is determined by $T_{k+1} = T_k/(1 + \beta T_k)$, where T_k is the temperature at the kth iteration and β is the cooling ratio. The final temperature, T_f, was set to 0.0001 and β varied to correspond to the appropriate number of solutions search for each problem size (see above).

3.1.3. Genetic algorithm. This is a generational GA with an initially random fixed population size of size 50. Selection of parents is done through 2-tournament selection. Uniform crossover is used, with probability of 0.60, and 50 matings are made, with an expected value of 30 children (0.60×50) during each generation, creating a potential population of size 50. All 50 potential population members are exposed to mutation with probability of 0.40. Mutation is identical to the move in SA, described above, with a probability of change of 10% per integer in the string. A mutated member replaces its original member in the potential population. The 50 members of the potential population are combined with the two best members from the current generation and ranked. The top 50 of the 52 are then chosen for the next generation, an elitist strategy.

3.2. Hybrid versions

Because of the well structured neighborhood of one link or node change, it seems useful to employ HC as part of SA or GA. It was hypothesized that this would improve the efficiency of the search by more quickly finding locally optimal solutions, especially for the large problems. Three straightforward versions were explored.

3.2.1. Seeded simulated annealing. The final solution obtained by HC was used as the initial solution in SA.

3.2.2. Seeded genetic algorithm. The final solution obtained by HC was included in the initial population.

3.2.3. Memetic algorithm. The term memetic algorithm (MA) comes from the notion of a meme, a unit that can be genetically modified by thought or experience (Dawkins, 1976). This is different from a gene, which is unaltered by experience. Radcliffe (1994) first formally defined a memetic algorithm as one that integrates local search as part of the reproductive mechanism. In this paper, HC is used during each generation to improve all solutions newly generated by crossover and mutation. In other words, after a new solution in GA is generated, it is altered to its local optimum through the HC routine described in Section 3.1.1 before being inserted into the population.

4. Computational experience

4.1. The test problems

Three test problems, ranging in search space size from 1.2×10^{19} to 5.1×10^{47}, were randomly generated on a 100 by 100 grid with 20, 35 and 50 nodes/links, respectively. Tables 1 and 2 show the specifications of the link and node alternatives. Three cost constraints were considered for each problem and these were computed using Eq. (7) for each test problem. This resulted in nine total problems and each algorithm was run ten times on each instance (with different random number seeds).

$$\begin{aligned} \textit{Lower Bound} &= \left[\sum_{j=1}^{L} c(3)d_j + (N * c(3)) \right] \\ \textit{Upper Bound} &= \left[\sum_{j=1}^{L} c(1)d_j + (N * c(1)) \right] \end{aligned} \tag{6}$$

Table 1. Alternatives for fiber optic cables (links).

Type	Description	Reliability	Unit cost[a]
1	SC to SC connector, multimode, PVC 62.5/125 duplex	0.9990	28
2	SC to ST connector, multimode, PVC 62.5/125 duplex	0.9950	17
3	ST to ST connector, multimode, PVC 62.5/125 duplex	0.9900	12

[a]Includes switch, router and connector costs.

Table 2. Alternatives for servers (nodes).

	Type		
Properties	1	2	3
Intel Pentium III Processor	2@1.4 GHz	2@1.4 GHz	1@1.4 GHz
Cache per Processor	512 KB	512 KB	512 KB
Memory	2 GB (2@1-GB DIMMS)	1 GB (2@512-MB DIMMS)	512 MB (2@256-MB DIMMS)
10000 RPM Ultra 160 SCSI Disk Drive	1 @ 36 GB	1 @ 72 GB	1 @ 36 GB
Reliability	0.9995	0.9909	0.9950
Cost	5300	4300	2800

Table 3. Results of HC, SA and GA (best for each problem instance shaded).

	HC			SA			GA		
Network[a]	Best	Average	C.V.	Best	Average	C.V.	Best	Average	C.V.
$G = (20, 20)$ C1	0.9318	0.9304	0.0010	0.9327	0.9326	0.0001	0.9325	0.9325	0
$G = (20, 20)$ C2	0.9679	0.9633	0.0028	0.9718	0.9717	0.0002	0.9718	0.9709	0.00119
$G = (20.20)$ C3	0.9839	0.9826	0.0018	0.9840	0.9840	0	0.9840	0.9840	0
$G = (35, 35)$ C1	0.8746	0.8722	0.0019	0.8763	0.8761	0.0004	0.8764	0.8763	6E-05
$G = (35, 35)$ C2	0.9337	0.9281	0.0040	0.9423	0.9412	0.0012	0.9426	0.9425	0.00019
$G = (35, 35)$ C3	0.9695	0.9690	0.0004	0.9699	0.9699	0	0.9699	0.9699	0
$G = (50, 50)$ C1	0.8199	0.8141	0.0050	0.8258	0.8251	0.0006	0.8254	0.8252	0.0002
$G = (50, 50)$ C2	0.8972	0.8917	0.0057	0.9141	0.9128	0.0011	0.9145	0.9139	0.0005
$G = (50, 50)$ C3	0.9547	0.9512	0.0048	0.9567	0.9567	0	0.9569	0.9569	0

[a]For tables, the network notation is (number of links, number of nodes), cost constraint level.

$D = \textit{Upper Bound} - \textit{Lower Bound}$

$$\begin{aligned} &\text{1st Cost Const.} = \textit{Lower Bound} + (0.25 * D) \\ &\text{2nd Cost Const.} = \textit{Lower Bound} + (0.50 * D) \\ &\text{3rd Cost Const.} = \textit{Lower Bound} + (0.75 * D) \end{aligned} \tag{7}$$

4.2. *Results of the metaheuristics*

Table 3 shows the results of the classic metaheuristics including the C.V. (coefficient of variation) over the ten seeds. GA returns better solutions than SA and HC for most problem instances and seeds. Both SA and GA dominated HC, as would be expected when comparing global optimization techniques against a locally optimum technique. When the C.V. columns are examined, it can be seen that variation of the results over the ten runs of GA is significantly smaller than that of SA or HC, and in fact, all runs of the GA are essentially equal, that is, there is almost no variation to seed.

Please note that while the improvements in reliability may seem small, recall that reliability is bounded by 1.0 and any increase in network reliability can result in significant performance improvements over the lifetime of the computer network.

4.3. *Hybrid versions*

This section gives results of combining HC with GA and SA. The maximum number of solutions considered in each hybrid was the same as in the non-hybrid versions except that MA was restricted to $\frac{1}{10}$th the number of solutions. Since the average CPU time per solution considered of the MA was 8.2 times that of the GA, this conservatively equalized

Table 4. Results of local search versions (best for each problem instance shaded).

	SA-HC			GA-HC			MA		
Network	Best	Average	C.V.	Best	Average	C.V.	Best	Average	C.V.
$G = (20, 20)$ C1	0.9327	0.9325	0.0001	0.9325	0.9325	0	0.9325	0.9316	0.0009
$G = (20, 20)$ C2	0.9718	0.9711	0.0009	0.9718	0.9717	0.0002	0.9718	0.9705	0.0012
$G = (20. 20)$ C3	0.9840	0.9840	0	0.9840	0.9840	0	0.9840	0.9840	0
$G = (35, 35)$ C1	0.8764	0.8761	0.0004	0.8764	0.8761	0.0002	0.8764	0.8762	0.0004
$G = (35, 35)$ C2	0.9415	0.9410	0.0007	0.9426	0.9425	0.0002	0.9426	0.9412	0.0011
$G = (35, 35)$ C3	0.9699	0.9699	0	0.9699	0.9699	0	0.9701	0.9701	0
$G = (50, 50)$ C1	0.8252	0.8250	0.00033	0.8254	0.8251	0.0002	0.8257	0.8254	0.0004
$G = (50, 50)$ C2	0.9138	0.9132	0.0004	0.9138	0.9136	0.0003	0.9145	0.9140	0.0006
$G = (50, 50)$ C3	0.9567	0.9566	6E-05	0.9569	0.9569	0	0.9569	0.9569	0

computational time. Table 4 shows the results. In comparisons of solution quality only (both mean and best of ten runs), there is little difference between SA and the seeded SA, and between the GA and the seeded GA. The seeded SA, as in the ordinary SA, performed less well in the larger problem instances. The MA performed best with equal or better solutions than either GA version in all but the first problem instance.

In the computational effort comparison, the chosen metric is again solutions searched (i.e., objective functions calculated). As mentioned earlier, in this class of problem, most of the computational effort is spent in the reliability calculation (Eq. (3)) and therefore it is most advantageous to reduce this as much as possible. Figure 2 shows the results (note the log scale on the y axis). The SA and its seeded version considered the most solutions, followed by the GA (both versions). The MA reduced solutions considered by over ten fold from the GA while HC had (as expected) the most modest computational requirements. Although computational effort increases about eight times per solution for the MA over the GA, there is still substantial improvement in overall computational effort for the MA as compared to the GA. As a further comparison, HC was performed as a multi-start for 50 times, which approximately equalized computational effort with the MA. Results are compared in Table 5. In all cases, 50 trials of HC still resulted in solutions of lower reliability than MA.

4.4. Actual network redesign problem

The backbone network of Gazi University in Ankara, Turkey was established in 1992 and is shown on a campus map in figure 3, with the eleven nodes listed in Table 6. Despite the many technological developments in the telecommunications area and also the increasing load on its backbone network, Gazi University has not made major changes to the existing structure since 1992. For this reason, the current network is unable to serve as a high performance network. The University administration has decided to renew all links and equipment in order

Table 5. Results for 50 runs of HC compared to MA.

Network	MA Best REL	HC Best REL
$G = (20, 20)$ C1	0.9325	0.9318
$G = (20, 20)$ C2	0.9718	0.9679
$G = (20.20)$ C3	0.9840	0.9839
$G = (35, 35)$ C1	0.8764	0.8746
$G = (35, 35)$ C2	0.9426	0.9337
$G = (35, 35)$ C3	0.9701	0.9695
$G = (50, 50)$ C1	0.8257	0.8199
$G = (50, 50)$ C2	0.9145	0.8972
$G = (50, 50)$ C3	0.9569	0.9547

to increase performance, considering both cost and reliability. The geographical locations of the facilities require a "star" network topology, which is known to have poor reliability. In order to help overcome this weakness, some of the closer facilities will be connected by additional links. The network has eleven nodes and thirteen links with distances given in Table 7.

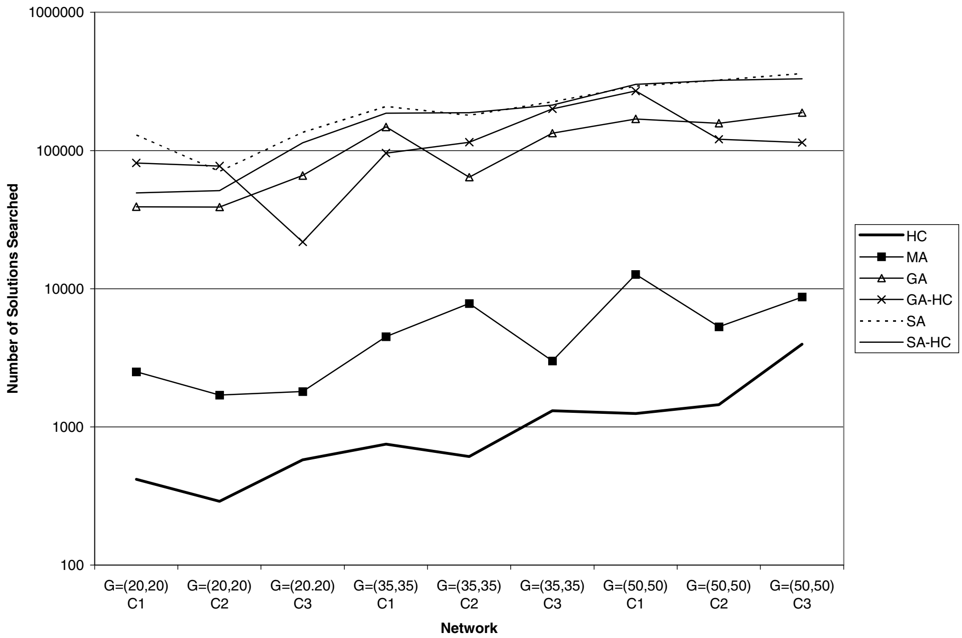

Figure 2. Average number of solutions searched for each heuristic approach.

Table 6. The entities in the Gazi University backbone network.

Entity	Node number
Rector (administration) building	1
Education	2
Science	3
Pharmacy	4
Technical education	5
Law	6
Dentistry	7
Medicine	8
Economics and administrative sciences	9
Engineering and architecture	10
Fine arts	11

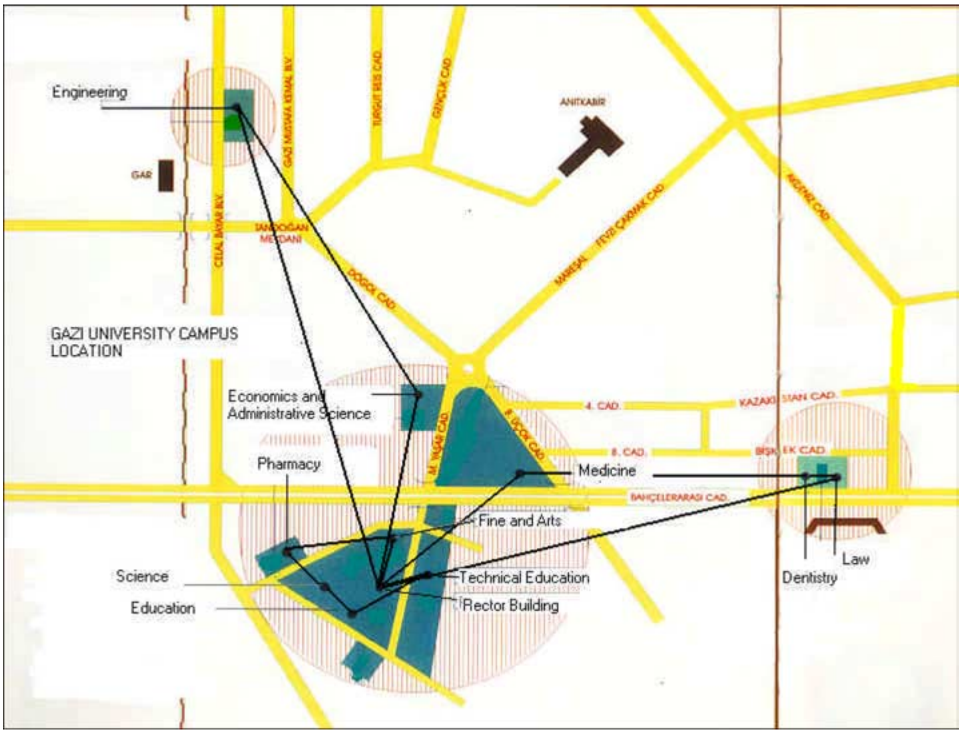

Figure 3. The topology of the Gazi University backbone network with the eleven node buildings and thirteen links marked.

Table 7. Links and their distances.

Links between entities (from Table 6)	Link number	Distance (meters)
1–5	1	220
1–6	2	2420
1–8	3	700
1–9	4	850
1–10	5	2300
1–11	6	120
2–3	7	150
2–5	8	350
3–4	9	200
4–11	10	450
6–7	11	150
7–8	12	1900
9–10	13	1500

Table 8. The recommended designs for each level of investment.

	Links													Nodes											
Investment	1	2	3	4	5	6	7	8	9	10	11	12	13	1	2	3	4	5	6	7	8	9	10	11	REL
\$218,635	1	3	2	2	3	1	1	2	1	3	1	3	3	1	1	1	1	1	1	1	1	1	1	1	0.9940
\$270,750	1	2	1	1	3	1	1	1	1	2	1	2	2	1	1	1	1	1	1	1	1	1	1	1	0.9944
\$322,865	1	2	1	1	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.9945

The University wanted to consider three levels of first costs to renovate this network. Using the MA, an objective function evaluation of Eq. (4) using the exact backtracking reliability calculation of Ball and Van Slyke (1977), and the link and node choices of Tables 1 and 2, the best designs for each cost level are shown in Table 8. While the dominant node choice is server type 1, links varied over the network. University officials can decide on the trade offs between investment and overall reliability in determining which of the three designs to implement.

5. Conclusions

This paper presented three classic metaheuristics—HC, SA and GA—and their hybrid forms to design computer communication networks with a fixed topology that have optimal overall reliability subject to a maximum cost constraint. The performance of these heuristics according to solution quality and computational effort was investigated and evaluated. HC, even when equalizing computational time, could not approach the best solutions of GA or MA, indicating that this problem class has many local optima. Thus, for better designs, a global heuristic must be employed. When SA and GA are compared according to solution quality, both performed well on the smaller instances, however GA exhibited less sensitivity to random number seed. However, GA performed better on the larger instances and needed substantially fewer function evaluations than SA overall.

Using local search to seed either the SA or the GA was not particularly advantageous. Using local search at each generation of the GA proved computationally important. In these cases, the MA provided equally good or better solutions as the other methods and substantially reduced function evaluations. Therefore, overall conclusions are that this problem class demands a global heuristic and that a GA is able to reliably exploit the structure through crossover and mutation. Using local search for new solutions created during evolution (the MA) was the most successful strategy. This provided not only the best designs for the larger problems, but also the lowest computational effort of the global heuristics. HC is a good method for a quick assessment of designs but cannot be relied upon to produced network designs of the highest quality.

Acknowledgment

Part of this study was funded by The Scientific and Technical Research Council of Turkey (TUBITAK) under grant MISAG-79, the National Planning Center of Turkey (DPT) under grant DPT-96K120820 and the U.S. National Science Foundation (NSF) under grant INT-9731207.

Note

1. Parallel links can be transformed into a single link using the link costs and reliabilities.

References

Aarts, E. and J.K. Lenstra. (1997). *Local Search in Combinatorial Optimization*. John Wiley & Sons.

Abuali, F.N., D.A. Schoenefeld, and R.L. Wainwright. (1994a). "Terminal Assignment in a Communications Network Using Genetic Algorithms." In *Proceedings of the ACM Computer Science Conference*, pp. 74–81.

Abuali, F.N., D.A. Schoenefeld, and R.L. Wainwright. (1994b). "Designing Telecommunications Networks Using Genetic Algorithms and Probabilistic Minimum Spanning Trees." In *Proceedings of the 1994 ACM Symposium on Applied Computing*, pp. 242–246.

Aggarwal, K.K., Y.C. Chopra, and J.S. Bajwa. (1982). "Topological Layout of Links for Optimizing the Overall Reliability in a Computer Communication System." *Microelectronics & Reliability* 22, 347–351.

Altiparmak, F., B. Dengiz, and A.E. Smith. (1998). "Reliability Optimization of Computer Communication Networks Using Genetic Algorithms." In *IEEE International Conference on System, Man and Cybernetics, SMC-98*, USA pp. 4676–4681.

Altiparmak, F., B. Dengiz, and A.E. Smith. (2000). "An Evolutionary Approach for Reliability Optimization in Fixed Topology Computer Networks." *Transactions on Operational Research in Turkey* 12, 57–75 (in English).

Atiqullah, M.M. and S.S. Rao. (1993). "Reliability Optimization of Communication Networks Using Simulated Annealing." *Microelectronics and Reliability* 33, 1303–1319.

Ball, M.O. and J.S. Provan. (1983). "Calculating Bounds on Reachability and Connectedness in Stochastic Networks." *Networks* 13, 253–278.

Ball, M. and R.M. Van Slyke. (1977). "Backtracking Algorithms for Network Reliability Analysis." *Annals of Discrete Mathematics* 1, 49–64.

Beltran, H.F. and D. Skorin-Kapov. (1994). "On Minimum Cost Isolated Failure Imune Networks." *Telecommunications Systems* 3, 183–200.

Brown, J.I., C.J. Colbourn, and J.S. Devitt. (1993). "Network Transformations and Bounding Network Reliability." *Networks* 23, 1–17.

Colbourn, C.J. (1987). *The Combinatorics of Network Reliability.* Oxford University Press, Oxford, U.K.

Davis, L., D. Orvosh, A. Cox, and Y. Qui. (1993). "A Genetic Algorithm for Survivable Network Design." In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 408–415.

Dawkins, R. (1976). *The Selfish Gene*. Oxford University Press, Oxford, U.K.

Deeter, D.L. and A.E. Smith. (1998). "Economic Design of Reliable Networks." *IIE Transactions* 30, 1161–1174.

DeJong, K.A. (1975). "An Analysis of the Behavior of a Class of Genetic Adaptive Systems." PhD Thesis, University of Michigan.

Dengiz, B., F. Altiparmak, and A.E. Smith. (1997a). "Local Search Genetic Algorithm for Optimal Design of Reliable Networks." *IEEE Transactions on Evolutionary Computation* 1, 179–188.

Dengiz, B., F. Altiparmak, and A.E. Smith. (1997b). "Efficient Optimization of All-Terminal Reliable Networks, Using an Evolutionary Approach." *IEEE Transactions on Reliability* 46, 18–26.

Eglese, R.W. (1990). "Simulated Annealing: A Tool for Operational Research." *European Journal of Operational Research* 46, 271–281.

Glover, F., M. Lee, and J. Ryan. (1991). "Least-Cost Network Topology Design for a New Service: An Application of a Tabu Search." *Annals of Operations Research* 33, 351–362.

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

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

Jan, R.H. (1993). "Design of Reliable Networks." *Computers and Operations Research* 20, 25–34.

Jan, R.H. F.J. Hwang, and S.T. Chen. (1993). "Topological Optimization of a Communication Network Subject to Reliability Constraint." *IEEE Transactions on Reliability* 42, 63–70.

Kirkpatrick, S., C.D. Gerlatt, and N.P. Vecchi. (1983). "Optimization by Simulated Annealing." *Science* 220, 671–680.

Koh, S.J. and C.Y. Lee. (1995). "A Tabu Search for the Survivable Fiber Optic Communication Network Design." *Computers & Industrial Engineering* 28, 689–700.

Koumolos, C., J.R. Antony, and R. Jean. (1994). "A Survey of Simulated Annealing Applications to Operations Research Problems." *Omega* 22, 41–56.

Kumar, A., R.M. Pathak, Y.P. Gupta, and H.R. Parsaei. (1995). "A Genetic Algorithm for Distributed System Topology Design." *Computers & Industrial Engineering* 28, 659–670.

Kumar, A., R.M. Pathak, and Y.P. Gupta. (1995). "Genetic Algorithm Based Reliability Optimization for Computer Network Expansion." *IEEE Transactions on Reliability* 44, 63–72.

Lundy, M. and A. Mees. (1986). "Convergence of an Annealing Algorithm." *Mathematical Programming* 34, 111–124.

Nel, L. and C.J. Colbourn. (1990). "Combining Monte Carlo Estimates and Bounds for Network Reliability." *Networks* 20, 277–298.

Pierre, S., M.-A. Hyppolite, J.-M. Bourjolly, and O. Dioume. (1995). "Topological Design of Computer Communication Networks Using Simulated Annealing." *Engineering Applications of Artificial Intelligence* 8, 61–69.

Pirlot, M. (1992). "General Local Search Heuristics in Combinatorial Optimization: A Tutorial." *Belgian Journal of Operations Research, Statistics and Computer Science* 32, 7–67.

Provan, J.S. and M.O. Ball. (1982). "The Complexity of Counting Cuts and of Computing the Probability that a Graph is Connected." *SIAM Journals on Computing* 12, 777–788.

Radcliffe, N.J. (1994). "Formal Memetic Algorithms." (T. Fogarty, *Evolutionary Computing* ed.), Springer Lecture Notes in Computer Science, vol. 865, pp. 250–263.

Smith, A. E. and D.W. Coit. (1997). "Constraint-Handling Techniques-Penalty Functions." in *Handbook of Evolutionary Computation*, Institute of Physics Publishing and Oxford University Press, Bristol, U.K., Chapter C5.2.

Srivaree-ratana, C., A. Konak, and A.E. Smith. (2002). "Estimation of All-Terminal Network Reliability Using An Artificial Neural Network." *Computers and Operations Research* 29, 849–868.

Venetsanopoulos, N. and I. Singh. (1986). "Topological Optimization of Communication Networks Subject to Reliability Constraints." *Problem of Control and Information Theory* 15, 63–78.

Yeh, M.-S., J.-S. Lin, and W.-C. Yeh. (1994). "A New Monte Carlo Method for Estimating Network Reliability." In *Proceedings of the 16th International Conference on Computers & Industrial Engineering*, pp. 723–726.