

Efficient Optimization of Reliable Two-Node Connected Networks: A Biobjective Approach

Abdullah Konak

Information Sciences and Technology, Penn State Berks, Reading, Pennsylvania 19610, konak@psu.edu

Alice E. Smith

Department of Industrial and Systems Engineering, Auburn University, Auburn, Alabama 36849, smithae@auburn.edu

This paper presents a biobjective genetic algorithm (GA) to design reliable two-node connected telecommunication networks. Because the exact calculation of the reliability of a network is NP-hard, network designers have been reluctant to use network reliability as a design criterion; however, it is clearly an important aspect. Herein, three methods of reliability assessment are developed: an exact reliability calculation method using factoring, an efficient Monte Carlo estimation procedure using the sequential construction technique and network reductions, and an upper bound for the all-terminal reliability of networks with arbitrary arc reliabilities. These three methods of reliability assessment are used collectively in a biobjective GA with specialized mutation operators that perturb solutions without disturbing two-node connectivity. Computational experiments show that the proposed approach is tractable and significantly improves upon the results found by single-objective heuristics.

Key words: networks; decision theory; multiple criteria; heuristics; reliability; genetic algorithms; simulation

History: Accepted by S. Raghavan, Area Editor for Telecommunications and Electronic Commerce; received February 2009; revised September 2009, May 2010; accepted June 2010. Published online in *Articles in Advance* August 31, 2010.

1. Introduction

Network reliability refers to the ability of a network to continue service in the case of component failures. Over the past decade, network reliability has become an important concern because of innovations in telecommunication technology and the increased dependence on services provided by telecommunication networks (Ball et al. 1995). The new technologies enabling capacity concentration, such as fiber-optic cables and ATM broadband technology, tend to engender sparse topologies consisting of a few high-capacity links. A sparse network, however, is highly vulnerable to component (links and nodes) failures as even a single component failure can render a network disconnected. Meanwhile, organizations and individuals are becoming more dependent on telecommunication networks to carry out their daily activities.

Providing connectivity is the primary function of a network. Therefore, the majority of work on network reliability has focused on connectivity-based measures. In connectivity-based analysis, a network with unreliable components is usually modeled by a *probabilistic network* $G = (E, V)$ with node set V and arc set E such that each arc $(i, j) \in E$ can be in either of the two states: operative or failed, with associated probabilities p_{ij} and $1 - p_{ij}$, respectively. Given a probabilistic network G with a specified node set

$K \subset V$, *K-terminal reliability* is defined as the probability that every node in K can communicate with every other node in K ; i.e., there exists at least one path between every pair of nodes in K . For backbone networks, all-terminal reliability, the probability that all network nodes are directly or indirectly connected (i.e., $K = V$), is the predominant reliability measure because these networks are expected to provide connectivity service for all possible node pairs. Several papers have addressed the network design problem considering all-terminal reliability (Jan 1993; Jan et al. 1993; Kumar et al. 1995; Dengiz et al. 1997a, b; Costamagna et al. 1998; Deeter and Smith 1998; AboElFotouh and Al-Sumait 2001; Srivaree-ratana et al. 2002; Kroese et al. 2007; Ramirez-Marquez and Rocco 2008; Watcharasitthiwat and Wardkein 2009).

An approach to ensure that a network remains operative after link and node failures is to impose redundant paths between communication centers so that connectivity can be achieved by alternative paths. In other words, the network is survivable in case of component failures. This approach leads to the network design problem known as the minimum cost *k-connected* network design problem. A network is said to be *k-arc connected* if at least k arcs must be removed to disconnect any two nodes of the network. Similarly, a network is *k-node connected* if at least

k nodes must be removed to disconnect the remaining nodes. For high-capacity fiber-optic networks, two-node/arc connectivity generally provides an adequate level of survivability (Grotschel et al. 1995, Monma and Shallcross 1989). Therefore, the majority of the work in network survivability focuses on designing minimum cost two-connected networks, such as in Davis et al. (1993), Dutta and Mitra (1993), Grötschel et al. (1995), Monma and Shallcross (1989), Pierre and Elgibaoui (1997), Pierre et al. (1995), Cheng (1998), Ghashghai and Rardin (2002), Mandal et al. (2003), and Chimani et al. (2007, 2008, 2010).

Connectivity measures such as two-node connectivity are often used as surrogates for reliability while designing networks because connectivity is much easier to calculate than reliability (Ball et al. 1995). Connectivity measures the survivability of a network; however, it does not necessarily correlate well with reliability. For example, consider the networks shown in Figures 1(a)–(c). In terms of connectivity, these networks are two-node connected. However, one additional arc, which does not improve the connectivity of the networks with respect to two-node connectivity, substantially improves the network reliability in (b) and (c). In addition, although networks (b) and (c) have similar network topologies (the same number of arcs and cycles), network (c) has a higher reliability as a result of the placement of the additional arc. As this example demonstrates, very similar topologies with respect to connectivity may have different reliability. Therefore, using a reliability metric in the network design process adds important information. However, the majority of work in network reliability focuses on the reliability analysis of a given network. A minority of them (Taka and Abe 1994; Costamagna et al. 1998; Deeter and Smith 1998; Dengiz et al. 1997a, b; Jan 1993a; Jan et al. 1993b; Kumar et al. 1995; Liu and Iwamura 2000; AboElFotouh and Al-Sumait 2001; Soni et al. 2004; Konak and Smith 2006; Kroese et al. 2007; Ramirez-Marquez and Rocco 2008; Watcharasitthiwat and Wardkein 2009) address the design problem.

In this paper, the biobjective network design problem is defined as follows. There are given a set of undirected arcs and L alternative technologies (arc types) that can be chosen for each arc. Let c_{ijl} and

p_{ijl} represent the cost and reliability of arc (i, j) with technology l , respectively. Select a subset of arcs with their arc types to construct a two-node connected network with minimum design cost and maximum all-terminal reliability. Unlike previous approaches to reliable network design, the problem in this paper is formulated as a biobjective problem with a connectivity constraint.

There are several motivations to casting the problem in this manner. Choosing a network topology is the first step of the network design process. Candidate topologies are refined in the detailed design phases based on the specifications and available technology. Because the quality of service of a network is much affected by component failures, the objective is designing a robust network topology for long-term network operation. All-terminal reliability is a good measure to gauge the robustness of alternative topologies against component failures and can be considered a strategic objective. Thus in topological design, cost and all-terminal reliability are not hard constraints but competing objectives. Another motivation for a biobjective approach is that in many real-life reliability optimization problems, component reliabilities are estimated based on historical observations. Because it is estimated, all-terminal reliability is better treated as an objective rather than a hard constraint. Finally, multiple superior design solutions are usually preferred to a single solution. This allows the decision maker to best determine the final design by weighing the trade-offs between cost and reliability among a nondominated set. Recently, several papers (Taboada et al. 2008; Kumar et al. 2009; Kim and Gen 2000; Huang et al. 2009, 2005; Azaron et al. 2009; Zhao et al. 2007; Konak et al. 2006) have addressed the need for multiobjective optimization approaches to reliable system design. The aim of this paper is to aid network designers in choosing a final network topology design by providing the trade-off curve of cost and reliability in terms of a set of network topologies that are not inferior to each other with respect to cost or reliability, that is, the Pareto set of network topologies.

The computational challenge of designing reliable networks is computing reliability. The exact calculation of most reliability measures is NP-hard (Ball 1980, Ball and Provan 1983). Therefore, a large body of network reliability research has addressed developing efficient techniques to evaluate network reliability, including exact methods, theoretical bounds, and simulation. Most theoretical bounds have been developed for specific cases, such as identical arc reliabilities, or they require complex calculations, limiting their usage for general networks. Therefore, simulation has been a popular alternative. However, using

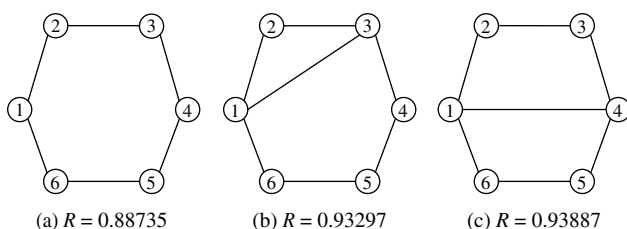


Figure 1 All-Terminal Reliability (R) and Network Topology ($p = 0.90$ for Each Link)

simulation to estimate reliability has drawbacks. Networks and network components are usually highly reliable systems. Therefore, a large number of replications are required to sample failed network states and make an accurate estimate, increasing the computational cost of simulation.

The first contribution of this paper is in the implementation of the genetic algorithm (GA). Standard GA crossover and mutation operators can yield infeasible and disconnected network topologies. The mutation operators of the biobjective GA in this paper maintain two-node connectivity and an efficient two-node connectivity check/repair algorithm is used in crossover.

The second contribution of this paper is the way that efficient computational techniques for reliability evaluation are incorporated into an optimization algorithm to reduce the computational burden of evaluating network reliability. A new upper bound on all-terminal reliability is developed for quick estimate of reliability. The upper bound is used to quickly eliminate unpromising solutions without further evaluating them. Using an algorithm based on the factoring theorem and network reductions (Page and Perry 1988), the exact reliability of networks much larger than previously reported is calculated. Finally, to complement the exact calculation, an efficient reduced variance simulation using network reductions is employed. The reliability evaluation procedure of the optimization approach exploits each reliability estimation technique. Computational experiments show that proposed approach is effective and yields superior results on a range of published test problems.

2. Description of the Optimization Method

2.1. Notation and Assumptions

This section describes the details of the optimization method. The notation is as follows.

| | |
|--------------------------------|---|
| $G = (V, E)$ | undirected network with node set V and arc set E . |
| n | number of nodes, $ V $. |
| m | number of arcs, $ E $. |
| (i, j) | undirected arc (i, j) between nodes i and j ($(i, j) = (j, i)$). |
| L | number of possible arc options. |
| $A = \{a_{ij}\}$ | solution A such that $a_{ij} = l$ if arc type l is used between nodes i and j . |
| $R(A)$ | all-terminal reliability of solution A . |
| $C(A)$ | cost of solution A . |
| μ | population size. |
| $\mu_{\max}, \mu_{\text{ini}}$ | maximum allowable and initial population size. |

| | |
|------------|--|
| P | population. |
| $cd(A)$ | crowding distance of solution A . |
| g | total number of solutions evaluated. |
| g_{\max} | maximum allowable levels of g to terminate the search. |

The following are the common notation and definitions used in the all-terminal reliability calculations. Specific notation is introduced as needed.

| | |
|--------------------|---|
| x_{ij} | the state of arc (i, j) such that $x_{ij} = 1$ if it is operative; $x_{ij} = 0$ otherwise. |
| \mathbf{x} | 0–1 state vector of the network, $\mathbf{x} = \{x_{ij}; (i, j) \in E\}$. |
| S | the space of all possible states of the network. |
| $\Phi(\mathbf{x})$ | the structure function of the network such that $\Phi(\mathbf{x}) = 1$ if all nodes are connected to each other in state \mathbf{x} ; $\Phi(\mathbf{x}) = 0$ otherwise. |
| p_{ij} | the reliability of arc (i, j) ($p_{ij} = p_{ji}$ and $p_{ij} = 0$ if arc (i, j) does not exist). |
| q_{ij} | the unreliability of arc (i, j) . |
| U | Uniform random variable between zero and one. |
| K | number of simulation replications. |

In the network reliability literature, the most common assumption is that arc failures are independent. Although in practice arc failures are often observed together, this assumption is imperative for computational tractability. Another assumption is perfectly reliable nodes (i.e., nodes do not fail). Finally, no repair is allowed in the model. Under these assumptions, the probability of observing a particular network state \mathbf{x} is given by

$$\Pr\{\mathbf{x}\} = \prod_{(i,j) \in E} [q_{ij} + x_{ij}(2p_{ij} - 1)], \quad (1)$$

and the all-terminal reliability of a network G is given by

$$R(G) = E[\Phi(\mathbf{x})] = \sum_{\mathbf{x} \in S} \Phi(\mathbf{x}) \Pr\{\mathbf{x}\}. \quad (2)$$

2.2. Multiobjective Genetic Algorithms

GA was first specified by Holland (1975), and since then it has been proven to be a useful global search technique for many difficult optimization problems. GA operates on a set of solutions called a *population*. GA has two operators to produce new solutions from existing solutions in the population: *crossover* and *mutation*. In crossover, generally two solutions, called parents, are combined together to produce new solutions, called *offspring*. The parents are selected from the population with preference toward better solutions so that offspring are expected to inherit good features of parents. By repeatedly applying crossover, features of good solutions are expected to appear more frequently in the population, eventually leading to convergence to an overall good solution. Mutation introduces random changes into characteristics of

solutions. Crossover leads the population to converge by encouraging the solutions in the population to become alike. Mutation reintroduces genetic diversity back into the population and helps the search avoid early convergence to local optima. In many combinatorial optimization problems including network design, the traditional GA crossover operators, such as single-point crossover and bit-flip mutation, can produce disconnected or infeasible network topologies. In this paper, uniform crossover with a repair algorithm to maintain two-node connectivity and mutation operators that always generate a two-node connected network are used.

A multiobjective decision problem is defined as follows: given an n -dimensional decision variable vector $\mathbf{x} = \{x_1, \dots, x_n\}$, find a vector \mathbf{x}^* that minimizes k objective functions $\mathbf{f}(\mathbf{x}^*) = \{f_1(\mathbf{x}^*), \dots, f_k(\mathbf{x}^*)\}$ subject to $g(\mathbf{x}^*) = \mathbf{b}$. In many real-life problems, the objectives under consideration conflict. Hence, optimizing \mathbf{x} with respect to a single objective often results in unacceptable results with respect to the other objectives. Therefore, a perfect multiobjective solution that simultaneously optimizes each objective function is almost impossible. A reasonable approach to a multiobjective problem is to investigate a set of solutions, each of which satisfies the objectives at an acceptable level without being dominated by any other solution. Assuming that all objectives are of the minimization type, a solution \mathbf{x} is said to dominate another solution \mathbf{y} if and only if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ for $i = 1, \dots, k$ and $f_j(\mathbf{x}) < f_j(\mathbf{y})$ for at least one objective j . A solution is said to be *Pareto-optimal* if it is not dominated by any other feasible solution in the solution space. A Pareto-optimal solution cannot be improved with respect to any objective without worsening at least one other objective. The set of all nondominated solutions is referred to as the *Pareto-optimal set*, and for a given Pareto-optimal set, the corresponding objective function values in the objective space are called the *Pareto front*. Interested readers should refer to the comprehensive survey papers on multiobjective GA by Fonseca and Fleming (1995), Deb (1999), Van Veldhuizen and Lamont (2000), Coello (2000), Tan et al. (2002), Konak et al. (2006), and Knowles and Nakayama (2008).

2.3. Problem Encoding and Initial Solutions

A node-adjacency matrix representation with arc types is used to represent solutions. In this representation, a network is stored in an $n \times n$ matrix, $A = \{a_{ij}\}$, such that $a_{ij} = l$ if arc type l is used between nodes i and j , $l = 0, \dots, L$ where type 1 denotes the least reliable, type L denotes the most reliable arc type, and type 0 means that no arc exists between nodes i and j . It should be noted that because matrix A is symmetric, only its upper half is adequate to represent solutions.

The GA is initialized with solutions randomly generated based on the two-tree heuristic given by Monma and Shallcross (1989) from a complete network. This procedure can quickly generate a great variety of two-node connected random networks. The first step of the two-tree heuristic is to find a random spanning tree of the nodes. In the second step, a second random spanning tree spanning only the leaves of the first spanning tree is added to the solution. After determining the topology, link types are randomly selected from integers 1 to L .

2.4. Uniform Crossover with Two-Node Connectivity Repair

The crossover operator has two parts: uniform crossover and a two-node connectivity check/repair algorithm. In uniform crossover, an offspring Z randomly inherits arcs from either of two parents X and Y , one at a time, as follows:

```

for  $i = 1, \dots, n, j = i + 1, \dots, n$  do
    if  $U > 0.5$  then  $z_{ij} = x_{ij}$  else  $z_{ij} = y_{ij}$ .
    
```

As a result of uniform crossover, it is possible that the offspring is not two-node connected or not even connected. Therefore, two-node connectivity of the offspring needs to be checked. If the offspring is not two-node connected, a repair algorithm is used to produce an acceptable topology. The procedure for the connectivity check/repair algorithm is given below.

Algorithm (Procedure connectivity check/repair)

```

1  for each node  $v \in V$  do {
2    remove node  $v$  from offspring  $Z$ 
3    let  $s$  be the node with the smallest index
      in  $V \setminus v$ 
4     $S(v) = \{s\}$ ,  $LIST = \{s\}$ , and  $\bar{S}(v) = V \setminus s$ 
5    while  $\bar{S}(v) \neq \emptyset$  do {
6      while  $LIST \neq \emptyset$  do {
7        select a node  $i$  in  $LIST$ 
8        if there exist a link  $(i, j)$  such that  $j \in \bar{S}(v)$ 
          then {
9          remove node  $j$  from  $\bar{S}(v)$  and add
            to  $S(v)$ 
10         add node  $j$  to the front of  $LIST$ 
11        }
12       else delete node  $i$  from  $LIST$ 
13      }
14     if  $\bar{S}(v) \neq \emptyset$  then {
15       select minimum distance link  $(i, j)$  such
         that  $i \in S(v)$ ,  $j \in \bar{S}(v)$ , and  $(i, j) \in X \cup Y$ 
16       remove node  $j$  from  $\bar{S}(v)$  and add to  $S(v)$ 
17       add node  $j$  to the front of  $LIST$ 
18     } } }
    
```

In essence, the algorithm is a depth-first search, which is invoked n times, after removing a single node v for $v = 1, \dots, n$ from the network, to check

the connectivity of the remaining nodes. The network is two-node connected if the remaining nodes are connected for $v = 1, \dots, n$. Otherwise, the network is repaired. In the procedure above, set $S(v)$ denotes the set of nodes that can be reached from the smallest indexed node after removing node v , and $\bar{S}(v)$ denotes the set of the nodes that cannot be reached. If $\bar{S}(v) = \emptyset$ when the depth-first search terminates, removing node v does not render the other nodes disconnected. If $\bar{S}(v) \neq \emptyset$, the remaining nodes are disconnected, and the network is repaired by adding the minimum distance arc (i, j) from the parents between the nodes in $S(v)$ and $\bar{S}(v)$, i.e., $i \in S(v)$ and $j \in \bar{S}(v)$. The type of the added arc is randomly determined. Then, the depth-first search continues from node j until two-node connectivity is achieved.

Using the repair algorithm, the two-node connectivity of a network can be determined in $O(nm)$ time. The depth-first search requires $O(m)$ time, and it is invoked n times, one for each node. If the offspring is not two-node connected, the repair operation requires only a few additional steps because the connectivity check and repairs are performed simultaneously. The connectivity check can be avoided in cases where one parent is a subset of the other. In this case, uniform crossover always produces a two-node connected offspring. Therefore, a quick check of this relationship between parents often prevented the connectivity check/repair.

2.5. Mutation Operators

The mutation operators are based on network perturbation heuristics developed by Monma and Shallcross (1989) to design minimum cost telecommunication

networks subject to two-node connected survivability constraints. These operators perturb a two-node connected network while preserving its two-node connectivity, eliminating the need for a connectivity check or repair. To achieve this, the mutation operators modify a network topology by replacing a cycle by another cycle. Given a certain number of links and assuming identical link reliabilities, the most reliable topologies include cycles. For example, using only n arcs, the most reliable network topology is a Hamilton cycle of n nodes. There is a strong relationship between cycles and network reliability. The mutation operators, which are described below, exploit this relationship to effectively search for highly reliable network topologies.

(a) *1-Arc, 2-Arc, and 3-Arc Exchange*: The first set of heuristics perturbs a solution without changing the number of arcs. Every two-node connected network includes at least a cycle, and replacing a cycle with another cycle of the same nodes does not disturb two-node connectivity (Monma and Shallcross 1989). The 2-arc exchange mutation operator finds a random cycle C in a solution, removes two randomly chosen arcs (a, b) and (c, d) of cycle C , and replaces them by arcs (a, c) and (b, d) , neither of which is in the solution, to obtain a new cycle C' (see Figure 2(a)). Similar to the 2-arc exchange mutation operator, the 3-arc exchange mutation operator transforms a cycle C to cycle C' by removing three links, (a, b) , (c, d) , and (e, f) , and adding three new links, (a, d) , (b, e) , and (c, f) . The 1-arc exchange mutation operator randomly changes the end nodes of an arc between the nodes of a randomly selected cycle C with the other nodes of the cycle (see Figure 2(b)). The added arc types are selected from the same types

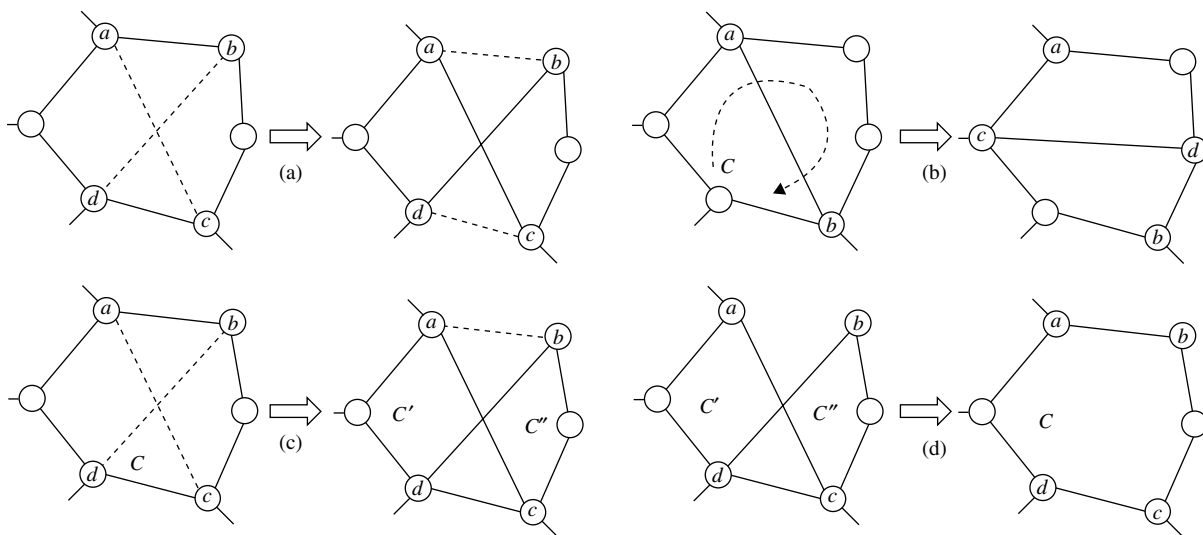


Figure 2 (a) The 2-Arc Exchange Mutation Operator; (b) the 1-Arc Exchange Mutation Operator; (c) the Pretzel Mutation Operator; (d) the Quetzal Mutation Operator

of the arcs removed. For example, in 2-arc exchange, if the arcs removed are types 1 and 3, one of the new arcs added is type 1 and the other one is type 3.

(b) *Pretzel and Quetzel Perturbation*: The second set of heuristics perturbs a solution by increasing or decreasing the number of arcs. The pretzel mutation operator removes a randomly selected arc (a, b) of a cycle C and adds two new arcs (a, c) and (b, d) as shown in Figure 2(c). Basically, the pretzel mutation operator splits a cycle C into two cycles C' and C'' without disturbing connectivity. The type of the new arcs is one step lower (less reliable) than the type of the removed arc (e.g., if a type 3 arc is removed, two type 2 arcs are added). The quetzel mutation operator is the opposite of the pretzel mutation; i.e., arc (a, b) is added to a solution while removing arcs (a, c) and (b, d) from the solution as shown in Figure 2(d). In other words, the quetzel mutation operator combines two cycles C' and C'' into a new cycle C , reducing the number of arcs by one while maintaining two-node connectivity. The type of new arc is one step higher (more reliable) than the maximum type of the removed arcs (e.g., if one type 1 and one type 2 arcs are removed, a type 3 arc is added).

(c) *Single-Arc Perturbation*: The previous mutation operators perturb a solution by changing at least two arcs of a solution. The one-arc perturbation heuristic changes a solution less radically. The *add-an-arc* mutation operator adds a nonexisting arc of random type to a solution. The *remove-an-arc* operator removes an arc without violating the connectivity constraint. To achieve this, a random cycle C is found and an arc (a, b) between the nodes of cycle C (excluding the ones on the cycle) is removed. For example, in Figure 2(b) arc (a, b) is removed. A final mutation operator, *change-arc-type*, is used to perturb arc types without changing the network topology by upgrading the type of an arc to the next higher type while degrading the type of another arc to the next lower type.

2.6. Population Update and Overall Algorithm

Figure 3 shows the overall procedure of the biobjective GA and reliability evaluation. Important features of the biobjective GA are summarized as follows:

- In each iteration only a single new solution is generated by either crossover or mutation, which is randomly determined with equal probabilities. For mutation, one of the eight mutation operators defined in §2.5 is also randomly selected with equal probabilities (change-arc-type mutation is not used if only one arc type exists).
- Population P (a subset of all nondominated solutions found so far) is maintained as sorted (i.e., $P = \{X_1, X_2, \dots, X_\mu\}$, where $C(X_i) \leq C(X_j)$ and $R(X_i) \leq R(X_j)$ for $i < j$).

- The crowding distance measure of nondominated sorting genetic algorithm-II (NSGA-II) (Deb et al. 2002) is used in the selection of solutions for crossover, mutation, and removal. The use of the crowding distance helps diversifying the search along the Pareto front. The crowding distance $cd(X)$ of solution X is a measure of how far solution X is located from its closest neighbors in a normalized objective function space. Because P is always sorted, the crowding distance of solution X_i is given as

$$cd(X_i) = \frac{C(X_{i+1}) - C(X_{i-1})}{C(X_\mu) - C(X_1)} + \frac{R(X_{i+1}) - R(X_{i-1})}{R(X_\mu) - R(X_1)} \quad (3)$$

for $i = 2, \dots, \mu - 1$ and $cd(X_1) = +\infty$ and $cd(X_\mu) = +\infty$.

- Restricted mating is applied in crossover. In the sorted population, solution X_i is only allowed to mate with the solutions in its neighborhood ($N(X_i)$), defined as

$$N(X_i) = \{X_j; \max\{1, i - a\} \leq j \leq \min\{\mu, i + a\}, i \neq j\}, \quad (4)$$

where a is an integer number defining the size of the neighborhood.

- The solutions for crossover and mutation are randomly selected using the crowding distance tournament selection (Deb et al. 2002). In this tournament selection, two solutions are randomly selected, and the one with the higher crowding distance is the winner.

As seen in Figure 3, a solution for mutation is selected using the crowding distance tournament selection, and then a mutation operator is randomly and uniformly selected among the mutation operators described previously. The crowding distance tournament selection is also used in the selection of the first parent for crossover. The second parent, however, is selected from the neighborhood of the first one.

If new solution Z is dominated by an existing solution X (i.e., $(C(X) \leq C(Z) \text{ and } R(X) > R(Z))$ or $C(X) < C(Z) \text{ and } R(X) \geq R(Z)$), solution Z is discarded. Similarly, if solution Z dominates an existing solution X , solution X is removed from the population. If new solution Z is nondominated, it is added to the population. Therefore, the population size is dynamic and it is possible that it might grow too unwieldy. In the biobjective GA, the size of the population (μ) is controlled by maximum population size (μ_{\max}) parameter. When $\mu > \mu_{\max}$, the solution with the smallest crowding distance is removed from the population. Each time the population is updated, the crowding distances of existing solutions are recalculated. Fortunately, this can be achieved very efficiently. When a new nondominated solution is found, it is inserted into the population while maintaining the ordering of the solutions. Therefore, the crowding distances are

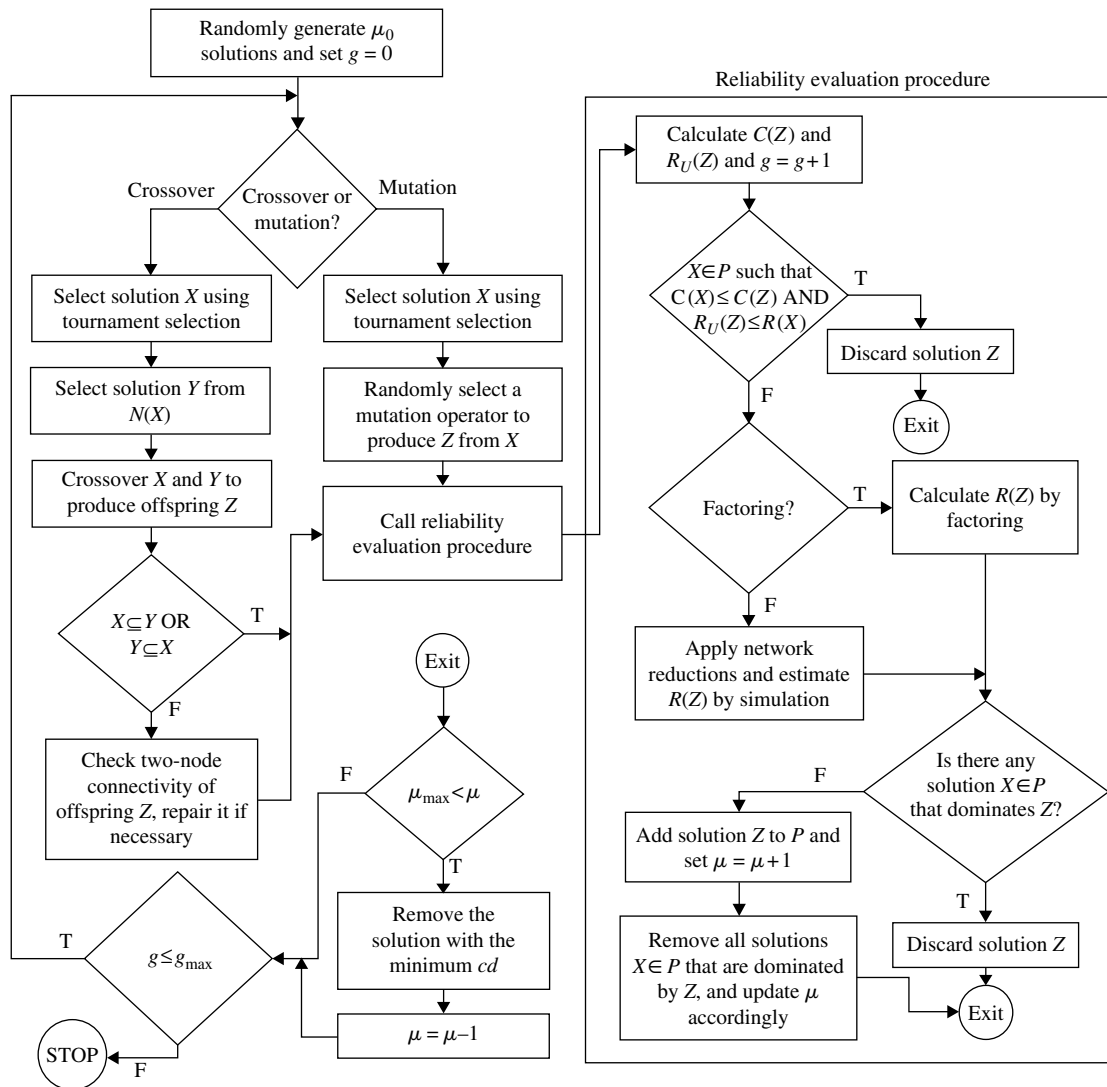


Figure 3 The Flow Chart for the Overall Procedure of the Biobjective GA and Reliability Evaluation Procedure

calculated only for the new solution and its immediate predecessor and successor. Similarly, when a solution is removed, the crowding distances of its immediate predecessor and successor are updated.

The proposed biobjective GA has similarities to NSGA-II (Deb et al. 2002) and Pareto-archived evolution strategy (PAES) (Knowles and Corne 1999). Like PAES, the biobjective GA creates one solution at a time, and the population and crowding distances are updated as soon as a new nondominated solution is identified. The main differences between PAES and the biobjective GA are that (1) crossover is used to generate new solutions, whereas PAES uses local search; (2) in PAES the current solution to which local search is applied is not updated until a better solution is found, whereas in the biobjective GA the solution for mutation is randomly selected each time; and (3) the crowding distance measure is used instead of the cell-based density approach of PAES to maintain

a diverse population. Relative to NSGA-II, the biobjective GA does not require a ranking scheme because only nondominated solutions are preserved in the population, and Pareto front updating is performed solution by solution rather than the generation batch updating of NSGA-II. In addition, a restricted mating is applied.

3. Evaluation of Reliability

This section presents the reliability evaluation procedure (see Figure 3). When a new solution, Z , is generated, its upper bound $R_U(Z)$ is computed and used to compare the new solution with the existing solutions in the population. If there exists a solution $X \in P$ such that $R(X) \geq R_U(Z)$ and $C(X) \leq C(Z)$, then solution Z is discarded. This initial check eliminates the need for an exact calculation or estimation by simulation for a clearly inferior solution. After the initial comparison,

if solution Z is not discarded, $R(X)$ is either exactly calculated or estimated by simulation, depending on network size and density.

Any exact technique to compute all-terminal reliability requires exponential time in the worst case. For undirected networks, factoring with network reductions provides the best possible time (see Ball et al. 1995 and Galtier et al. 2005 for a comprehensive discussion on the exact methods). The factoring code used in this paper to compute $R(X)$ follows the procedure given by Page and Perry (1988). If a network is too large for exact calculation of reliability by factoring, simulation with network reductions is used. To choose between factoring and simulation, the size and density of the network is considered. After network reliability calculation or estimation, new solution Z is once more compared with the existing solutions and the population is updated accordingly.

3.1. A New Upper Bound

Different lower and upper bounding approaches have been proposed for network reliability. These bounds can be grouped as (1) bounds based on the reliability polynomial (such as Ball and Provan 1983, Van Slyke and Frank 1972) or (2) bounds based on arc packing by cut or path sets (such as Brecht and Colbourn 1988, Lomonosov and Polesskii 1972, Sharafat and Ma'rouzi 2009). The bounds in the first group count operational network states and obtain bounds by counting a fraction of all possible states. A notable shortcoming of these bounds is that they are only applicable to networks with identical arc reliabilities. The second group of bounds considers a fraction of all possible cut/path sets of a network to obtain a bound. These bounds can be used for networks with arbitrary arc reliabilities, but they are computationally intensive because they require generating cut/path sets or an effective arc packing of the network.

In this paper, an efficient upper bound that can accommodate nonidentical arc reliabilities is developed. It is inspired by the use of bounds for fathoming nodes in branch and bound. The bound is

$$R(G) \leq 1 - \left(\sum_{i=1}^n \left(\left(\prod_{(i,k) \in E_i} (1 - p_{ik}) \right) \cdot \prod_{j=1}^{i-1} \left(1 - (1 - p_{ij})^{-1} \prod_{(j,k) \in E_j} (1 - p_{jk}) \right) \right) \right), \quad (5)$$

where E_i is the set of arcs incident to node i . This has a worst case complexity of $O(n^3)$ because only cut sets separating individual nodes are considered. The derivation of the upper bound is given in Online Supplement A (available at <http://joc.pubs.informs.org/ecompanion.html>).

3.2. Simulation

Several simulation approaches (e.g., Kumamoto et al. 1980, Van Slyke and Frank 1972, Easton and Wong 1980, Elperin et al. 1991, Fishman 1986, Kumamoto et al. 1977, Konak 2009) have been proposed in the literature. A new implementation of the sequential construction (SC) technique, originally proposed by Easton and Wong (1980) and later improved by Fishman (1986), is used within the search algorithm. The SC technique starts with all arcs in the failed state, and they are sequentially repaired one at a time. Let Π be a permutation of arcs representing the order in which they are repaired, and let $x_{[r]}$ represent the state of the r th arc in permutation. In the SC technique, the estimator of $R(G)$ for a randomly sampled permutation Π is given as Ball et al. (1995):

$$\hat{R}(G) = \frac{\sum_{j=0}^m \Phi(\mathbf{x}(\Pi; j)) \Pr\{\mathbf{x}(\Pi; j)\}}{\sum_{j=0}^m \Pr\{\mathbf{x}(\Pi; j)\}}, \quad (6)$$

where $\mathbf{x}(\Pi; j)$ is the state of the network such that $x_{[r]} = 1$ for $r = 1, \dots, j$ and $x_{[r]} = 0$ for $r = j + 1, \dots, m$, and $\Phi(\mathbf{x}(\Pi; j)) = 1$ if all nodes are connected in state $\mathbf{x}(\Pi; j)$, and 0 otherwise. In Online Supplement B, the detailed implementation of the SC technique in this paper is given.

3.3. Combining Simulation and Network Reductions

In network reductions, a network G is reduced to a network G' with fewer nodes and/or arcs such that

$$R(G) = \lambda R(G'), \quad (7)$$

where λ is a reliability-preserving multiplicative constant depending on the reductions. Network reductions provide substantial performance improvement during factoring by eliminating unnecessary pivoting and by reducing the size of a network. In this paper, network reductions are applied before using simulation to reduce the variance of the reliability estimation. Only two-degree reduction and parallel arc reductions are considered because networks must satisfy two-node connectivity, which requires that nodes must have at least two incident arcs.

In two-degree reduction, each node i connected to the network by only two arcs (i, j) and (i, k) is deleted from the network along with its arcs, and a new arc (j, k) with $p_{jk} = p_{ij}p_{ik}/(p_{ij} + p_{ik} - p_{ij}p_{ik})$ is added to the network and $\lambda = (p_{ij} + p_{ik} - p_{ij}p_{ik})$, i.e., the probability that at least one of arcs (i, j) and (i, k) is operative. In parallel arc reduction, two parallel arcs (i_1, j_1) and (i_2, j_2) between nodes i and j are replaced by a single arc (i, j) with $p_{ij} = 1 - (1 - p_{i_1j_1})(1 - p_{i_2j_2})$ and $\lambda = 1$. Parallel arcs might occur during merging nodes in two-degree reduction.

Given a network G , two-degree reductions and, if required, parallel arc reductions, are repeatedly applied until the reduced network G' has no nodes with degree 2 and no parallel arcs. After applying k successful reductions, the final λ is calculated as $\lambda = \prod_{i=1, \dots, k} \lambda_i$, where λ_i is the multiplicative constant of the i th reduction. Then, the SC technique is used to estimate $R(G')$, and $R(G)$ is estimated by

$$\hat{R}(G) = \lambda \hat{R}(G'). \quad (8)$$

The analysis of variance reduction resulting from network reductions is difficult to study for the SC technique. However, if crude Monte Carlo (CMC) sampling is used with network reductions, the variance reduction factor resulting from network reductions is given by

$$\Delta\sigma^2 = \frac{1 - R(G)}{\lambda - R(G)}. \quad (9)$$

The proof of Equation (9) is given in Online Supplement C. Equation (9) suggests that the variance reduction resulting from network reductions is a function of $\lambda - R(G)$. Therefore, any reduction applied to highly reliable networks is expected to provide much higher variance reductions than reductions to less reliable networks. This result is also intuitive. Network reductions can be applied only to sparsely connected nodes of a network, which, in fact, are highly dominant in the reliability expression of the network. With these nodes removed, $R(G')$ becomes almost equal to 1, which in turn makes $\lambda - R(G)$ almost 0. Therefore, significant variance reduction can be achieved for reducible dense networks. Computational gains by applying network reductions before simulation are presented in §4.3.

4. Computational Experience

The list of the test problems are given in Table 1. The first set includes the 8-, 9-, 10-, 15-, 20-, and 25-node test problems (set I) given in Dengiz et al. (1997a). These problems have identical arc reliabilities (i.e., no arc choice is available); therefore, the decision variables are binary. In Dengiz et al. (1997b), a single-objective GA is used to design networks when minimizing cost given an all-terminal reliability constraint and the node degree constraint of 2 as a minimum survivability requirement. In addition, a branch-and-bound algorithm is used to find optimal solutions for small problem instances. The second set of problems from Deeter and Smith (1998) includes the 10-, 14-, and 19-node test problems (set II) considering three arc choices with different reliabilities. Allowing multiple choices of arc type for each possible connection complicates the application of GA to the problem as a binary encoding is no longer applicable and

Table 1 Test Problems and Mean Computational Effort

| Problem set | n | p | CPU seconds |
|-------------|-----|---------------------|-------------|
| I-1 | 8 | 0.90 | 166 |
| I-2 | 8 | 0.95 | 166 |
| I-3 | 9 | 0.90 | 908 |
| I-4 | 9 | 0.95 | 903 |
| I-5 | 10 | 0.90 | 6,344 |
| I-6 | 10 | 0.95 | 6,560 |
| I-7 | 15 | 0.90 | 701 |
| I-8 | 20 | 0.95 | 1,048 |
| I-9 | 25 | 0.95 | 1,811 |
| II-1 | 10 | (0.70, 0.80, 0.90) | 669 |
| II-2 | 14 | (0.70, 0.80, 0.90) | 996 |
| II-3 | 19 | (0.96, 0.975, 0.99) | 986 |
| III-1 | 30 | 0.95–0.99 | 1,669 |
| III-2 | 40 | 0.95–0.99 | 1,877 |

exponentially increases the search space. In Deeter and Smith (1998), a single-objective GA is used to solve two versions of the problem without considering a connectivity constraint: (1) minimizing cost given a reliability constraint, and (2) maximizing reliability given a cost constraint. The third set of problems (set III) includes larger test problems, 30- and 40-node networks, to gauge scale-up. These problems are significantly larger than reliable network design problems studied previously. The x and y coordinates of the nodes are randomly generated on a 100×100 grid. The cost of each link is equal to the Euclidean distance between its nodes. The link reliabilities are set between 0.95 and 0.99 based on the length of links, 0.99 being the reliability of the shortest link and 0.95 being of the longest link with the others linearly scaled between.

While evaluating the reliability for the networks with 10 nodes or fewer, factoring was used regardless of the network density. For the networks larger than 10 nodes and fewer than 30 nodes, factoring is used if $m \leq 1.5n$; otherwise, the SC method with network reductions was used. For the 40-node problem, the selection criterion between factoring and simulation is $m \leq 1.2n$. For all runs, the parameters of the biobjective GA were $K = 7,500$ simulation replications, maximum population size $\mu_{\max} = 100$, initial population size $\mu_0 = 2$, and neighborhood size $a = 10$.

4.1. Comparing with Other Approaches

The biobjective GA was compared to NSGA-II (Deb et al. 2002) and PAES (Knowles and Corne 1999). The NSGA-II algorithm was implemented as defined in Deb et al. (2002) while using the problem-specific crossover and the mutation operators introduced in §2 (the 1-arc, 2-arc, and 3-arc exchange; pretzel; quetzal; add-and-arc; and remove-an-arc mutation operators). In mutation, one of the seven mutation operators is randomly selected with equal probabilities. Both the

biobjective GA of this paper and NSGA-II were initialized the same and used the same mutation rate (0.50) and the same population size. In other words, all aspects other than the search method were identical.

PAES is a multiobjective optimization algorithm that is based on the (1 + 1) local-search evolution strategy. The implementation of PAES in this paper uses the same mutation operators of the biobjective GA as local-search operators. Similarly, a mutation operator is randomly selected with equal probabilities. To achieve a diverse final Pareto front in PAES, the objective space is divided into K -dimensional cells that are dynamically updated when new maximum and minimum values of the objective functions are identified. The number of solutions in each cell is counted, and this information is used as a tie breaker between the current solution and the offspring solution. If both are nondominated, the one residing in the less crowded cell is selected as the winner. In the implementation of PAES in this paper, $K = 10$, and the maximum population size for PAES was 100.

In multiobjective optimization, two sets P_A and P_B of nondominated solutions found by two different multiobjective algorithms A and B are usually compared using the coverage measure (Wu and Azarm 2001) as follows:

$$C(P_A, P_B) = \frac{|\{Y \in P_B; \exists X \in P_A: X \succ Y\}|}{|P_B|}, \quad (10)$$

which gives the ratio of the solutions in set P_B that are dominated by at least one solution in set P_A . It should be noted that $C(P_A, P_B)$ and $C(P_B, P_A)$ are not complementary; i.e., $C(P_A, P_B) \neq 1 - C(P_B, P_A)$. Therefore, when comparing two multiobjective algorithms A and B , both $C(P_A, P_B)$ and $C(P_B, P_A)$ should be provided.

The coverage measure does not address the range of objective function values. The overall Pareto spread (OS) measure can be used to compare two sets P_A and P_B (Wu and Azarm 2001) with respect to the range of solutions. For the problem defined in this paper, the OS measure is calculated as

$$OS(P_A, P_B) = \frac{(\max_{X \in P_A} (R(X)) - \min_{X \in P_A} (R(X))) (\max_{X \in P_A} (C(X)) - \min_{X \in P_A} (C(X)))}{(\max_{X \in P_B} (R(X)) - \min_{X \in P_B} (R(X))) (\max_{X \in P_B} (C(X)) - \min_{X \in P_B} (C(X)))}$$

If $OS(P_A, P_B) > 1$, set P_A is preferred to set P_B with respect to the overall spread.

We used both coverage and overall spread measures to compare the biobjective GA with NSGA-II and PAES. For each test problem, 10 random replications were performed, and then the 10 sets of nondominated solutions found by the biobjective GA were compared to the 10 sets found by the other algorithms one at a time (i.e., for each algorithm pair,

100 comparisons were performed). Then, an overall coverage value was calculated by averaging individual values of the 100 comparisons as follows:

$$\bar{C}(P_A, P_B) = \frac{\sum_{i=1}^{10} \sum_{j=1}^{10} C(P_{A,i}, P_{B,j})}{100},$$

where $P_{A,i}$ denotes the set of nondominated solutions found in the i th replication of algorithm A . Similarly, an overall OS measure is calculated as follows:

$$\overline{OS}(P_A, P_B) = \frac{\sum_{i=1}^{10} \sum_{j=1}^{10} OS(P_{A,i}, P_{B,j})}{100}.$$

In addition, $\bar{C}(P_A, P_B)$ and $\overline{OS}(P_A, P_B)$ values were calculated for 50 different levels of stopping criteria from 1,000 to 50,000 with 1,000 increments. Therefore, the convergence of each algorithm relative to the others could be observed and the comparisons became independent from the stopping criteria parameter.

Figure 4 gives the plots of $\bar{C}(P_{\text{bi-GA}}, P_{\text{NSGA}})$, denoted by hollow squares; $\bar{C}(P_{\text{NSGA}}, P_{\text{bi-GA}})$, denoted by solid circles; and $\overline{OS}(P_{\text{bi-GA}}, P_{\text{NSGA}})$, denoted by a continuous line over different levels of the stopping criteria parameter for some of the test problems. The name of axes is omitted in the individual plots for the sake of figure clarity. As seen in the plots, NSGA-II dominates the biobjective GA at the beginning of the search because NSGA-II starts the search with 100 random solutions, whereas the biobjective GA starts with two solutions. In the majority of the test problems, more than 80% of the final nondominated solutions found by NSGA-II were dominated by the solutions found by the biobjective GA. With respect to the overall spread, the biobjective GA also performed better. In every case, $\overline{OS}(P_{\text{bi-GA}}, P_{\text{NSGA}})$ is greater than one at the termination. Both algorithms explore the same or very similar extreme points as the search progresses, but the biobjective GA identified them sooner. Figure 5 gives the plots of $\bar{C}(P_{\text{bi-GA}}, P_{\text{PAES}})$, $\bar{C}(P_{\text{PAES}}, P_{\text{bi-GA}})$, and $\overline{OS}(P_{\text{bi-GA}}, P_{\text{PAES}})$. It is clear that the biobjective GA is superior to PAES throughout the search in every test problem.

4.2. Comparing with Results from the Literature

Table 2 summarizes the results found for the 8-, 9-, and 10-node problems of set I by the biobjective GA and the optimal results found by a branch-and-bound algorithm and the best GA results of Dengiz et al. (1997b). For all but the smallest problem, the biobjective GA found the reported optimal solutions. For these smaller problems, Ramirez-Marquez and Rocco (2008) finds optimal solutions and AboElFotouh and Al-Sumait (2001) finds optimal solutions for all but problem I-3. In Figure 6, the Pareto fronts found for problems I-7, I-8, and I-9 are shown comparatively with the results found by the GA approach of Dengiz

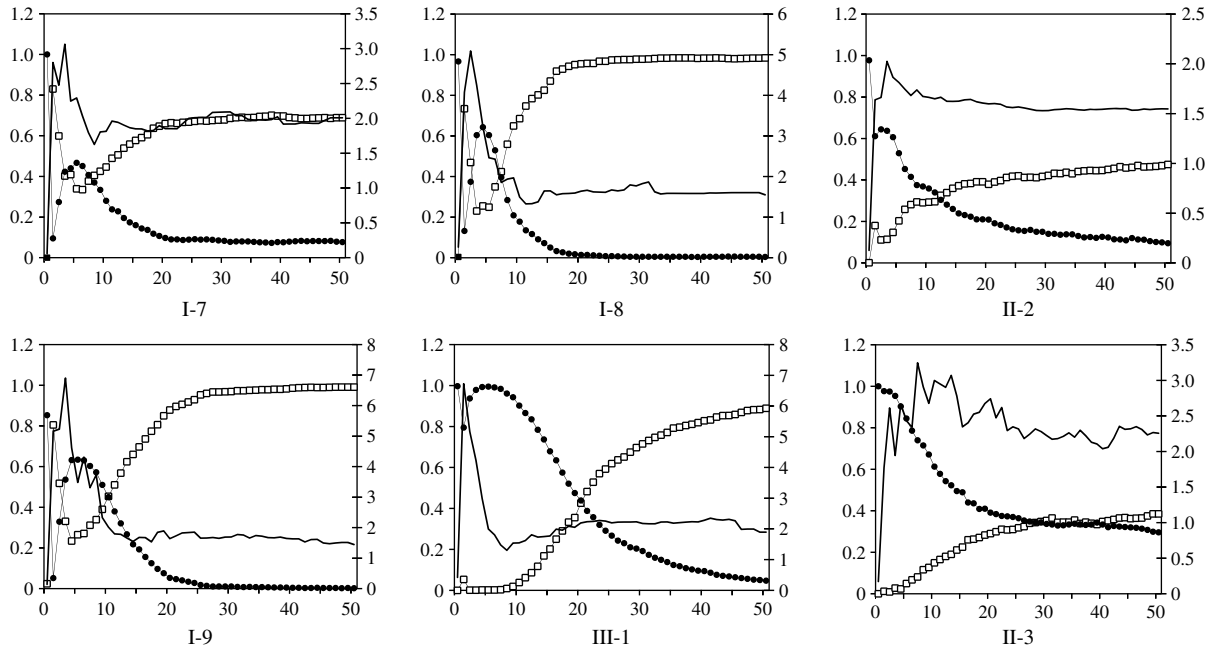


Figure 4 Comparison of the Biobjective GA and NSGA-II on Test Problems

Notes. x Axis: number of solutions evaluated (in 1,000s). Left y axis: overall coverage measure [\square , $\bar{C}(P_{\text{Bi-GA}}, P_{\text{NSGA}})$]; \bullet , $\bar{C}(P_{\text{NSGA}}, P_{\text{Bi-GA}})$]; right y axis: overall spread measure [$-$, $\overline{OS}(P_{\text{Bi-GA}}, P_{\text{NSGA}})$].

et al. (1997b), the artificial neural network (ANN) approach of AboElFotouh and Al-Sumait (2001), and the probabilistic search algorithm (PSA) of Ramirez-Marquez and Rocco (2008) (in all figures, the top

left corner is the ideal considering both objectives). For the larger problems, the biobjective GA found a large number of solutions dominating the best solutions reported in Dengiz et al. (1997b), AboElFotouh

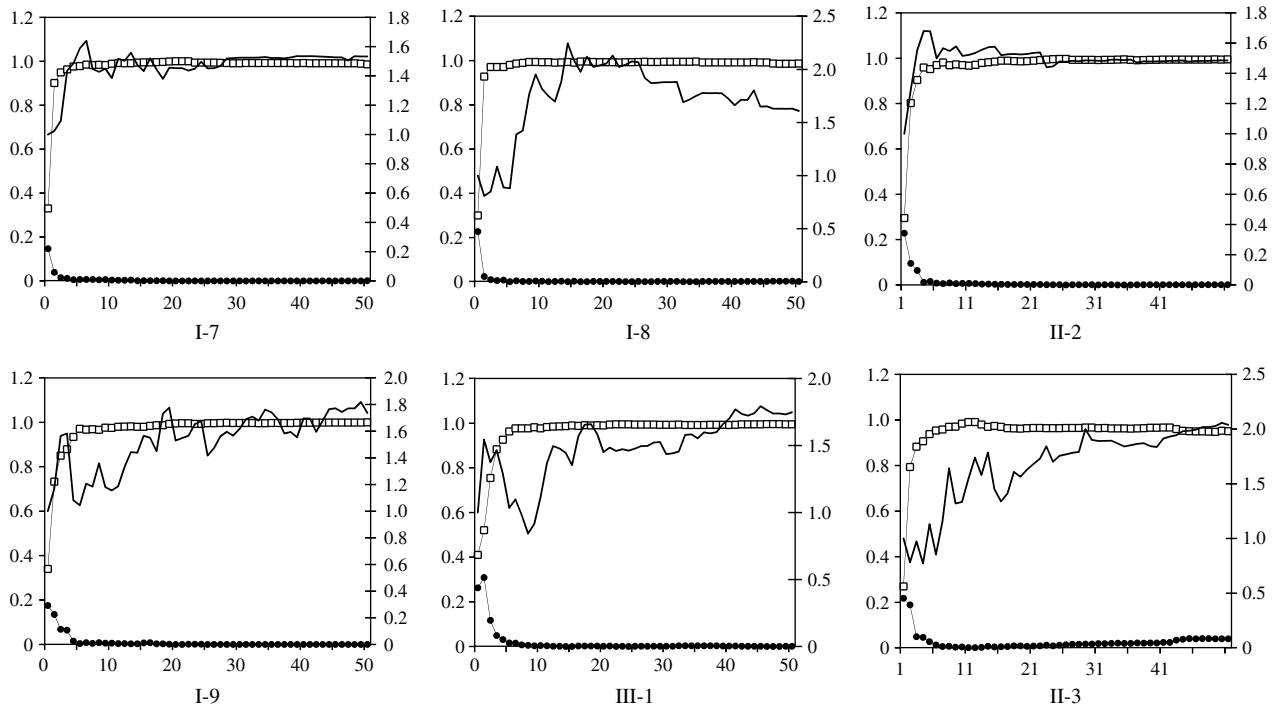


Figure 5 Comparison of the Biobjective GA and PAES on Test Problems

Notes. x Axis: number of solutions evaluated (in 1,000s). Left y axis: overall coverage measure [\square , $\bar{C}(P_{\text{Bi-GA}}, P_{\text{PAES}})$]; \bullet , $\bar{C}(P_{\text{PAES}}, P_{\text{Bi-GA}})$]; right y axis: overall spread measure [$-$, $\overline{OS}(P_{\text{Bi-GA}}, P_{\text{PAES}})$].

Table 2 Results for the Reliable Network Design Test Problems

| No. | n | p | Dengiz et al. (1997b) results for min $C(X)$, given $R(G)$ | Optimal $C(X)$ reported for the problems | Our results that dominate Dengiz et al. (1997b) |
|-----|-----|------|---|--|---|
| I-1 | 8 | 0.90 | (203, 0.90) | 203 | — |
| I-1 | 8 | 0.90 | (247, 0.95) | 247 | (247, 0.961376) |
| I-2 | 8 | 0.95 | (179, 0.95) | 179 | — |
| I-3 | 9 | 0.90 | (239, 0.90) | 239 | (239, 0.906564) |
| I-3 | 9 | 0.90 | (286, 0.95) | 286 | (286, 0.956670) |
| I-4 | 9 | 0.95 | (209, 0.95) | 209 | (209, 0.966935) |
| I-5 | 10 | 0.90 | (156, 0.90) | 154 | (154, 0.905014) (197, 0.951644) |
| I-5 | 10 | 0.90 | (205, 0.95) | 197 | (199, 0.952535) |
| I-6 | 10 | 0.95 | (136, 0.95) | 136 | (136, 0.961130) |

and Al-Sumait (2001), and Ramirez-Marquez and Rocco (2008). These results are noteworthy because the previous results are found by single-objective approaches that can intensively search a specified region defined by constraints, whereas the biobjective GA must search and identify a wider range of solutions.

As seen in Figure 6, the biobjective GA identified solutions spread along the Pareto front. When identical arc reliabilities are used, the lowest cost Pareto set solution is the optimal traveling salesman problem (TSP) tour of nodes, and the most reliable Pareto set solution is the fully connected network. In Figure 6, the solutions corresponding to the optimal TSP

tours are also shown. For the 15-node problem (I-7), the final GA population includes the optimal TSP tour. For the 20-node and 25-node problems, the final GA population does not include the solutions corresponding to the optimal TSP tours but the Pareto set approaches this theoretical solution with respect to cost. For instance, the optimal TSP tour for the 25-node problem has a cost and reliability of 238 and 0.64. The closest solution found by the biobjective GA has a cost and reliability of 256 and 0.88, respectively. At the high cost end of the Pareto front, many solutions with reliability of 1.0 (to the precision kept) were found. These solutions are very dense, but they are not fully connected. Note that generally the solutions at the extreme ends of the Pareto front are not practical and the decision maker would normally specify limits on cost and reliability. In the figures, the cutoff point (termed the factoring and simulation line) for using factoring or simulation to evaluate $R(G)$ is shown. For solutions below the cutoff point, $R(G)$ is exactly calculated by factoring. The preponderance of solutions in the Pareto set had their reliability calculated exactly.

Similar results were obtained for the set II problems with different arc choices. For each test problem, the final population included many solutions dominating the best solutions reported in Deeter and Smith (1998), and a wide and diverse Pareto front was found, as shown in Figure 7.

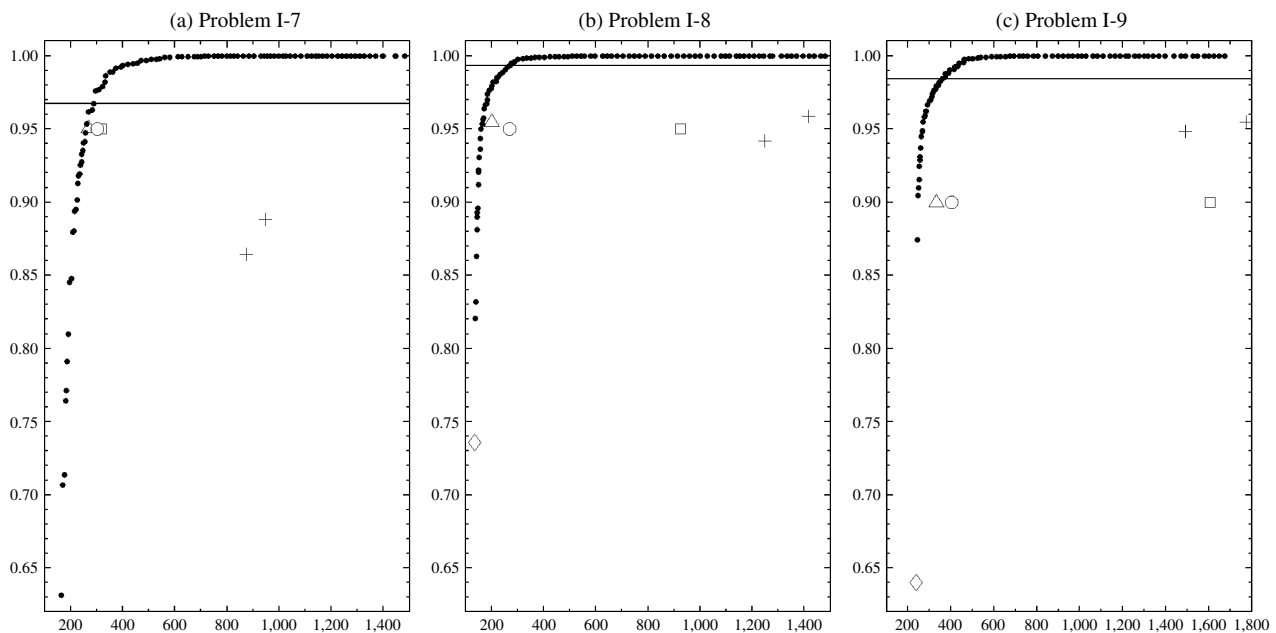


Figure 6 Comparison of the Biobjective GA with Respect to the Previously Published Results for (a) I-7, (b) I-8, and (c) I-9

Notes. x Axis: cost; y axis: all-terminal reliability [□, previously reported solution from Dengiz et al. (1997b); ○, previously reported solution from AboEIFotoh and Al-Sumait (2001); △, previously reported solution from Ramirez-Marquez and Rocco (2008); ●, final nondominated set found by the biobjective GA]. +, starting solutions; ◇, the minimum cost TSP solution; —, factoring and simulation line.

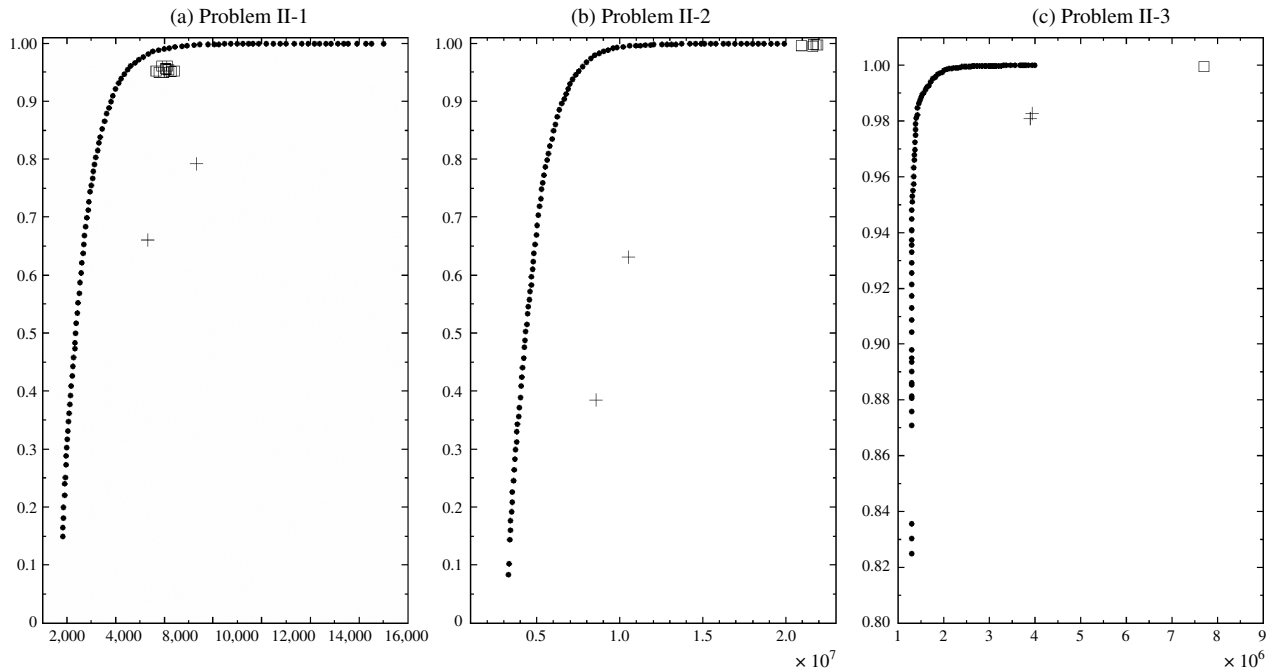


Figure 7 Comparison of the Biobjective GA with Respect to the Previously Published Results for (a) II-1, (b) II-2, and (c) II-3
 Notes. x Axis: cost; y axis: all-terminal reliability [□, previously reported solutions in Deeter and Smith (1998); ●, final nondominated set found by the biobjective GA]. +, starting solutions.

4.3. Computational Efficiency and Accuracy of Reliability Estimation

As mentioned earlier, the computational challenge of considering reliability in network design is computing reliability. In this paper, we combine the new upper bound, the SC simulation technique with network reductions, and factoring to balance precision with computational parsimony. The first step in reliability evaluation, the upper bound, significantly contributed to this goal as shown in Figure 8. The upper bound check eliminated the majority of inferior solutions, and its effectiveness slightly improved as the population converged. Moreover, the upper bound was most effective for dense networks (eliminating 60%–80% of

them) that are computationally more difficult to evaluate using simulation or factoring. Figure 9 shows the precision of the new bound relative to Jan’s bound (Jan 1993) and actual reliability. Note that the new bound is also more general than Jan’s in that arcs of differing reliability are allowed.

To test the efficiency of the SC simulation technique with network reductions, a set of experiments were performed using randomly generated 10-node networks with identical arc reliabilities. These networks ranged in number of arcs from 13 to 30, and 30 networks were randomly generated for each level of network density. Experiments were repeated for three different arc reliability levels, $p = 0.90, 0.95, \text{ and } 0.99$.

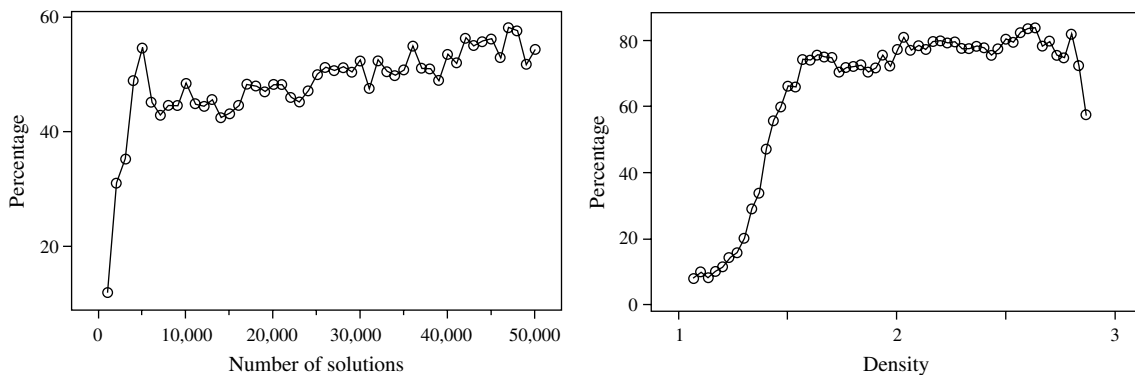


Figure 8 Percentage of the Solutions Eliminated by the Upper-Bound Check vs. Network Density
 Notes. The ratio of the number of arc to the number of nodes and search progress for problem III-1. Similar results were observed for other problems.

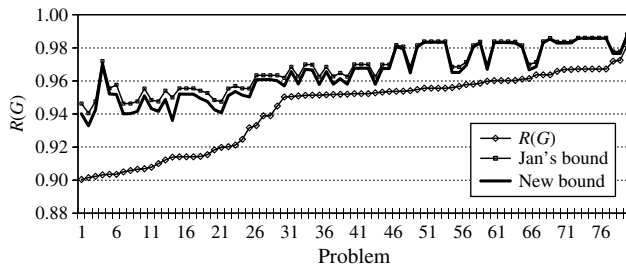


Figure 9 The Comparison of the Proposed Bound with Jan's Bound (1993) and the Actual Reliability for the 79 Networks Given in Dengiz et al. (1997b)

In all cases, $K = 10^6$ was used, and $R(G)$ was calculated exactly for each network using factoring, which enabled calculating the theoretical variance for crude Monte Carlo simulation.

Table 3 summarizes the variance reductions achieved using the SC simulation with network reductions. The first part of the table, σ_A^2/σ_B^2 , is the variance reductions resulting from network reductions only. The results in this part were obtained as follows. First, the reliability of each random network was estimated using the SC method. Let σ_B^2 be the variance of the estimations before network reductions. Then, network reductions were applied, and reliability was estimated again using the SC method. Let σ_A^2 be the variance of the estimation after applying reductions. Therefore, the variance reduction factors σ_B^2/σ_A^2 in the table are independent from the variance reductions due to the SC method. In the second part of the table, the ratio of the theoretical variance of CMC to σ_A^2 is given. This ratio indicates the combined effect of the SC method and network reductions in reducing the sampling error. The table shows that this combined method greatly reduces variance, which allows for fewer simulation replications. When the median variance reductions for 30 random networks were analyzed considering network density, the greatest variance reductions were achieved for two cases: very sparse networks and very dense networks. Sparse networks usually have several nodes to apply network reductions, and as a result, a sparse network is able to be greatly simplified through network reductions. In the case of dense networks, only a few reductions can be applied. However, these few reductions provide very high levels of reduction in

variance. Note that $1 > \lambda > R(G)$ and $R(G') > R(G)$ if a network G is reducible through network reductions, and a dense network is usually highly reliable (i.e., $R(G) \cong 1.0$). Therefore, $\lambda - R(G)$ in Equation (9) becomes very small in the case of a reducible dense network and in turn results in a large reduction in the variance. Because the majority of dense network needs to be evaluated by simulation, the proposed reliability evaluation scheme is very effective.

Table 1 summarizes the CPU times to solve the problems for $g_{\max} = 50,000$. The 30- and 40-node problems are the largest problem instances that have been studied in literature considering reliability. Considering the size of the problems, the reported CPU times are very acceptable.

5. Conclusions and Discussion

Network reliability and survivability have become important considerations in network design because of increased dependence on services provided by telecommunication networks. Exacerbating this is that recent innovations in telecommunication technology have resulted in sparser networks, where overall reliability is more dependent on the links' placement and individual reliability. However, because the exact calculation of reliability is NP-hard, practitioners have been reluctant to use reliability in the design process. In this paper, a biobjective GA is proposed to solve the reliable network design problem considering a survivability constraint.

Special attention is given to the efficient estimation and exact calculation of reliability. A new upper bound on reliability is developed that is general and can be used for networks with different arc reliabilities. This bound is effective in avoiding unnecessary evaluations of reliability, significantly increasing the computational reasonableness of the approach. The idea of applying network reductions before simulation is also a significant contribution of this paper, enhancing the practicality of accurately estimating network reliability through simulation for highly reliable networks. The analysis of variance reduction by network reduction presented in this paper is far from exhaustive. However, the initial experiments and the basic theoretical analysis have shown that even a few simple reductions can significantly increase the accuracy of simulation. It should be noted that it is not

Table 3 Minimum, Median, and Maximum Variance Reduction Achieved for 30 Random Networks

| | σ_B^2/σ_A^2 | | | $\sigma_{R(G)}^2/\sigma_A^2$ | | |
|------|-------------------------|-----------|-------------------|------------------------------|------------|-------------------|
| | Min | Median | Max | Min | Median | Max |
| 0.90 | 4 | 185 | 89,340 | 18 | 449 | 193,175 |
| 0.95 | 10 | 1,995 | 6,705,195 | 83 | 7,039 | 25,583,911 |
| 0.99 | 202 | 1,041,845 | 1,147,275,074,272 | 14,484 | 10,361,703 | 5,600,347,736,444 |

always possible to apply network reductions nor is it always computationally feasible to exactly calculate reliability, especially for dense networks. However, these are more theoretical concerns than practical ones. In reality, most telecommunications backbone networks are sparse enough that exact methods or simulation with network reduction can be applied. Of course, the newly developed bound and the innovation of using network reduction prior to simulation can be used independent of the particular optimization method herein.

There are not many results in the literature with which to form comparisons because larger network problems have avoided reliability as an objective or constraint because of the computational concerns mentioned. Our experiments show that the proposed approach can effectively search over the Pareto front and even improve on solutions found by single objective optimization approaches. The mutation operators using cycles to perturb a solution without disturbing two-node connectivity contribute to the bicriteria GA's success. Because the biobjective GA explores topologies made of cycles, it is particularly effective for the design problem posed. As further research, the computational performance of simulation–optimization approaches like the biobjective GA herein may be further improved by using hashing methods to quickly detect whether a solution has been previously investigated or not.

References

- AboElFotouh, H. M. F., L. S. Al-Sumait. 2001. A neural approach to topological optimization of communication networks, with reliability constraints. *IEEE Trans. Reliability* 50(4) 397–408.
- Azaron, A., C. Perkgöz, H. Katagiri, K. Kato, M. Sakawa. 2009. Multi-objective reliability optimization for dissimilar-unit cold-standby systems using a genetic algorithm. *Comput. Oper. Res.* 36(5) 1562–1571.
- Ball, M. O. 1980. Complexity of network reliability computations. *Networks* 10(2) 153–165.
- Ball, M. O., J. S. Provan. 1983. Calculating bounds on reachability and connectedness in stochastic networks. *Networks* 13(2) 253–278.
- Ball, M. O., C. J. Colbourn, J. S. Provan. 1995. Network reliability. M. Ball, T. L. Magnanti, C. L. Monma, G. L. Nemhauser, eds. *Network Models: Handbooks in Operations Research and Management Science*, Vol. 7. Elsevier, Amsterdam, 673–745.
- Brecht, T. B., C. J. Colbourn. 1988. Lower bounds on two-terminal network reliability. *Discrete Appl. Math.* 21(3) 185–198.
- Cheng, S.-T. 1998. Topological optimization of a reliable communication network. *IEEE Trans. Reliability* 47(3) 225–233.
- Chimani, M., M. Kandyba, P. Mutzel. 2007. A new ILP formulation for 2-root-connected prize-collecting Steiner networks. *Algorithms—ESA 2007, Proc. 15th Eur. Sympos.* Lecture Notes in Computer Science, Vol. 4698. Springer-Verlag, Berlin, 681–692.
- Chimani, M., M. Kandyba, I. Ljubić, P. Mutzel. 2008. Strong formulations for 2-node-connected Steiner network problems. B. Yang, D.-Z. Du, C. Wang, eds. *Combin. Optim. Appl. Second Internat. Conf. (COCO A 2008)*. Lecture Notes in Computer Science, Vol. 5165. Springer-Verlag, Berlin, 190–200.
- Chimani, M., M. Kandyba, I. Ljubić, P. Mutzel. 2010. Orientation-based models for $\{0, 1, 2\}$ -survivable network design. *Math. Programming Ser. B* 124(1–2) 413–440.
- Coello, C. A. C. 2000. An updated survey of GA-based multiobjective optimization techniques. *ACM Comput. Surveys* 32(2) 109–143.
- Costamagna, E., A. Fanni, G. Giacinto. 1998. A tabu search algorithm for the optimisation of telecommunication networks. *Eur. J. Oper. Res.* 106(2–3) 357–372.
- Davis, L., D. Orvosh, A. Cox, Y. Qiu. 1993. A genetic algorithm for survivable network design. *Proc. Fifth Internat. Conf. Genetic Algorithms, Urbana-Champaign, IL*, Morgan Kaufmann, San Francisco, 408–415.
- Deb, K. 1999. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *J. Evolutionary Comput.* 7(3) 205–230.
- Deb, K. D., A. Pratap, S. Agarwal, T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary Comput.* 6(2) 182–197.
- Deeter, D. L., A. E. Smith. 1998. Economic design of reliable networks. *IIE Trans.* 30(12) 1161–1174.
- Dengiz, B., F. Altıparmak, A. E. Smith. 1997a. Efficient optimization of all-terminal reliable networks, using an evolutionary approach. *IEEE Trans. Reliability* 46(1) 18–26.
- Dengiz, B., F. Altıparmak, A. E. Smith. 1997b. Local search genetic algorithm for optimal design of reliable networks. *IEEE Trans. Evolutionary Comput.* 1(3) 179–188.
- Dutta, A., S. Mitra. 1993. Integrating heuristic knowledge and optimization models for communication network design. *IEEE Trans. Knowledge Data Engrg.* 5(6) 999–1017.
- Easton, M. C., C. K. Wong. 1980. Sequential destruction method for Monte Carlo evaluation of system reliability. *IEEE Trans. Reliability* R-29(1) 27–32.
- Elperin, T., I. Gertsbakh, M. Lomonosov. 1991. Estimation of network reliability using graph evolution models. *IEEE Trans. Reliability* 40(5) 572–581.
- Fishman, G. S. 1986. A comparison of four Monte Carlo methods for estimating the probability of $s - t$ connectedness. *IEEE Trans. Reliability* 35(2) 145–155.
- Fonseca, C. M., P. J. Fleming. 1995. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Comput.* 3(1) 1–16.
- Galtier, J., A. Laugier, P. Pons. 2005. Algorithms to evaluate the reliability of a network. *Fifth Internat. Workshop Design Reliable Comm. Networks, Naples, Italy*, 93–100.
- Ghashghai, E., R. L. Rardin. 2002. Using a hybrid of exact and genetic algorithms to design survivable networks. *Comput. Oper. Res.* 29(1) 53–66.
- Grotschel, M., C. L. Monma, M. Stoer. 1995. Polyhedral and computational investigations for designing communication networks with high survivability requirements. *Oper. Res.* 43(6) 1012–1024.
- Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.
- Huang, H.-Z., J. Qu, M. Zuo. 2009. Genetic-algorithm-based optimal apportionment of reliability and redundancy under multiple objectives. *IIE Trans.* 41(4) 287–298.
- Huang, H.-Z., Z. Tan, M. J. Zuo. 2005. Intelligent interactive multi-objective optimization method and its application to reliability optimization. *IIE Trans.* 37(11) 983–993.
- Jan, R.-H. 1993. Design of reliable networks. *Comput. Oper. Res.* 20(1) 25–34.
- Jan, R.-H., F.-J. Hwang, S.-T. Chen. 1993. Topological optimization of a communication network subject to a reliability constraint. *IEEE Trans. Reliability* 42(1) 63–70.

- Kim, J. R., M. Gen. 2000. A genetic algorithm for bicriteria communication network topology design. *Engrg. Valuation Cost Anal.* 3 351–363.
- Knowles, J., D. Corne. 1999. The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation. *Proc. IEEE Congress on Evolutionary Comput. (CEC 99)*, IEEE, Washington, DC, 98–105.
- Knowles, J., H. Nakayama. 2008. Meta-modeling in multiobjective optimization. J. Branke, K. Deb, K. Miettinen, R. Stowinski, eds. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Lecture Notes in Computer Science, Vol. 5252. Springer-Verlag, Berlin, 245–284.
- Konak, A. 2009. Efficient event-driven simulation approaches to analysis of network reliability and performability. *Internat. J. Model. Simulation* 29(2) 156–168.
- Konak, A., A. E. Smith. 2006. Network reliability optimization. M. G. C. Resende, P. M. Pardalos, eds. *Handbook of Optimization in Telecommunications*. Springer, New York, 735–760.
- Konak, A., D. W. Coit, A. E. Smith. 2006. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engrg. System Safety* 91(9) 992–1007.
- Kroese, D. P., K.-P. Hui, S. Nariai. 2007. Network reliability optimization via the cross-entropy method. *IEEE Trans. Reliability* 56(2) 275–287.
- Kumamoto, H., K. Tanaka, K. Inoue. 1977. Efficient evaluation of system reliability by Monte Carlo method. *IEEE Trans. Reliability* R-26(5) 311–315.
- Kumamoto, H., K. Tanaka, K. Inoue, E. J. Henley. 1980. Dagger-sampling Monte Carlo for system unavailability evaluation. *IEEE Trans. Reliability* R-29(2) 122–125.
- Kumar, R., K. Izui, M. Yoshimura, S. Nishiwaki. 2009. Multi-objective hierarchical genetic algorithms for multilevel redundancy allocation optimization. *Reliability Engrg. System Safety* 94(4) 891–904.
- Kumar, A., M. R. Pathak, P. Y. Gupta, H. R. Parsaei. 1995. A genetic algorithm for distributed system topology design. *Comput. Oper. Res.* 28(3) 659–670.
- Liu, B., K. Iwamura. 2000. Topological optimization models for communication network with multiple reliability goals. *Comput. Math. Appl.* 39(7–8) 59–69.
- Lomonosov, M. V., V. P. Poleskii. 1972. Lower bound of network reliability. *Problemy Peredachi Informatsii* 8(2) 118–123.
- Mandal, S., D. Saha, R. Mukherjee, A. Roy. 2003. An efficient algorithm for designing optimal backbone topology for a communication networks. *Proc. ICCT 2003–Internat. Conf. Comm. Technology, Beijing*, Vol. 1, 103–106.
- Monma, C. L., D. F. Shallcross. 1989. Methods for designing communications networks with certain two-connected survivability constraints. *Oper. Res.* 37(4) 531–541.
- Page, L. B., J. E. Perry. 1988. A practical implementation of the factoring theorem for network reliability. *IEEE Trans. Reliability* 37(3) 259–267.
- Pierre, S., A. Elgibaoui. 1997. A tabu-search approach for designing computer-network topologies with unreliable components. *IEEE Trans. Reliability* 46(3) 350–359.
- Pierre, S., M.-A. Hyppolite, J.-M. Bourjolly, O. Dioume. 1995. Topological design of computer communication networks using simulated annealing. *Engrg. Appl. Artificial Intelligence* 8(1) 61–69.
- Ramirez-Marquez, J. E., C. M. Rocco. 2008. All-terminal network reliability optimization via probabilistic solution discovery. *Reliability Engrg. System Safety* 93(11) 1689–1697.
- Sharafat, A. R., O. R. Ma’rouzi. 2009. All-terminal network reliability using recursive truncation algorithm. *IEEE Trans. Reliability* 58(2) 338–347.
- Soni, S., S. Narasimhan, L. J. LeBlanc. 2004. Telecommunication access network design with reliability constraints. *IEEE Trans. Reliability* 53(4) 532–541.
- Srivaree-ratana, C., A. Konak, A. E. Smith. 2002. Estimation of all-terminal network reliability using an artificial neural network. *Comput. Oper. Res.* 29(7) 849–868.
- Taboada, H. A., J. F. Espiritu, D. W. Coit. 2008. MOMS-GA: A multi-objective multi-state genetic algorithm for system reliability optimization design problems. *IEEE Trans. Reliability* 57(1) 182–191.
- Taka, M., T. Abe. 1994. Network reliability design techniques to improve customer satisfaction. *IEEE Comm. Magazine* 32(10) 64–68.
- Tan, K. C., T. H. Lee, E. F. Khor. 2002. Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. *Artificial Intelligence Rev.* 17(4) 253–290.
- Van Slyke, R., H. Frank. 1972. Network reliability analysis I. *Networks* 1(3) 279–290.
- Van Veldhuizen, D. A., G. B. Lamont. 2000. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Comput.* 8(2) 125–147.
- Watcharasitthiwat, K., P. Wardkein. 2009. Reliability optimization of topology communication network design using an improved ant colony optimization. *Comput. Electrical Engrg.* 35(5) 730–747.
- Wu, J., S. Azarm. 2001. Metrics for quality assessment of a multi-objective design optimization solution set. *Trans. J. Mechanical Design* 123(1) 18–25.
- Zhao, J.-H., Z. Liu, M.-T. Dao. 2007. Reliability optimization using multiobjective ant colony system approaches. *Reliability Engrg. System Safety* 92(1) 109–120.