Multicriteria Network Design Using Evolutionary Algorithm

Rajeev Kumar and Nilanjan Banerjee

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur, 721 302, India
rkumar@cse.iitkgp.ernet.in

Abstract. In this paper, we revisit a general class of multi-criteria multi-constrained network design problems and attempt to solve, in a novel way, with Evolutionary Algorithms (EAs). A major challenge to solving such problems is to capture possibly all the (representative) equivalent and diverse solutions. In this work, we formulate, without loss of generality, a bi-criteria bi- constrained communication network topological design problem. Two of the primary objectives to be optimized are network delay and cost subject to satisfaction of reliability and flow-constraints. This is a *NP-hard* problem so we use a hybrid approach (for initialization of the population) along with EA. Furthermore, the two-objective optimal solution front is not known *a priori*. Therefore, we use a multiobjective EA which produces diverse solution space and monitors convergence; the EA has been demonstrated to work effectively across complex problems of *unknown* nature. We tested this approach for designing networks of different sizes and found that the approach scales well with larger networks. Results thus obtained are compared with those obtained by two traditional approaches namely, the exhaustive search and branch exchange heuristics.

1 Introduction

Network design problems where even a single cost function or objective value (e.g., minimal spanning tree or shortest path problem) is optimized, are often NP-hard [1]. Many such uni-criterion network design problems are well studied and many heuristics/methods exist for obtaining exact/approximate solutions in polynomial-time [2]. But, in most real-life applications, network design problems generally require simultaneous optimization of multiple and often conflicting objectives, subject to satisfaction of some constraints. For example, topological design of communication networks, particularly mesh/wide area networks is a typical multiobjective problem involving simultaneous optimization of cost of the network and various performance criteria such as average delay of the network, throughput subject to some reliability measures and bandwidth/flow-constraints. The problem can be stated as: given a set of node locations and the traffic between the nodes, it is required to design the layout of links between the

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 2179–2190, 2003.

nodes while optimizing certain criteria e.g., overall cost, average per packet delay, reliability and provision for expansion. This requires optimization of conflicting factors, subject to various constraints. For example, reducing the packet delay could mean an increase in the link capacities, which will result in an increase in the network cost. Exploring the whole solution space for such a design problem is an NP hard problem [3]. Similar design problems exist for multicast routing of multimedia communication in constructing a minimal cost spanning/Steiner tree with given constraints on diameters [4].

Such multicriteria network design problems occur in many other engineering applications too. In VLSI design, the interconnect resistance increases significantly with deep micron technology. An increase in interconnect resistance increases interconnect delays thus making a dominant factor in timing analysis of VLSI circuits. The VLSI circuit design aims at finding minimum cost spanning/Steiner tree given delay bound constraints on source-sink connections [5]. Analogously, there exists the problem of degree/diameter- constrained minimum cost networks [6].

Many NP-hard bicriteria network design problems have been attempted and approximate solutions obtained using heuristics/methods, and verified in polynomial time, see - [6], [7] and [8]. For example, Ravi et al. [8] and Deo et al. [6] presented approximation algorithm by optimizing one criterion subject to a budget on the other. We argue that the use of heuristics may yield *single* optimized solutions in each objective-space, and may not yield many other equivalent solutions. Secondly, extending this approach to multi-criteria problems (involving more than two objectives/constraints) the techniques require improving upon more than one constraints. Thirdly and more importantly, such approaches may not yield all the representative optimal solutions. Most conventional approaches to solve network design problems start with a Minimum Spanning Tree (MST), and thus effectively minimizes the cost. With some variations induced by ϵ-constraint method, most other solutions obtained are located near the minimal-cost region of the Pareto-front, and thus do not form the complete Pareto-front.

In this work, we try to overcome the disadvantages of conventional techniques and single objective EAs. We use multiobjective EA to obtain a Pareto-front. For a wide-ranging review and a critical analysis of evolutionary approaches to multiobjective optimization - see [9] and [10]. There are many implementation of multiobjective EAs, for example, MOGA [11], NSGA [12], SPEA [13]) and PEAS [14]. These implementations achieve diverse and equivalent solutions by some diversity preserving mechanism, they do not talk about convergence. Any explicit diversity preserving method needs prior knowledge of many parameters and the efficacy of such a mechanism depends on successful fine-tuning of these parameters. In a recent study, Purshouse & Fleming [17] extensively studied the effect of sharing, along with elitism and ranking, and concluded that while sharing can be beneficial, it can also prove surprisingly *ineffective* if the parameters are not carefully tuned.

Some other recent studies have been done on combining convergence with diversity. Laumanns et al. [15] proposed an ϵ-dominance for getting an ϵ-approximate Pareto-front for problems whose optimal Pareto-set is *known*. Kumar & Rockett [16] proposed use of Rank-histograms for monitoring convergence of Pareto-front while maintaining diversity without any *explicit* diversity preserving operator. Their algorithm is demonstrated to work for problems of *unknown* nature. Secondly, assessing convergence does not need any *a priori* knowledge for monitoring movement of Pareto-front using rank-histograms.

In this work, we use their Pareto Converging Genetic Algorithm (PCGA) [16] which has been demonstrated to work effectively across complex problems and achieves diversity without needing *a priori* knowledge of the solution space. PCGA excludes any explicit mechanism to preserve diversity and allows a natural selection process to maintain diversity. Thus multiple, equally good solutions to the problem, are provided. PCGA assesses convergence to the Pareto-front which, by definition, is unknown in most real search problems, by use of rank-histograms.

We select topological design of communication network as a sample network problem domain. We present a novel approach to design a network with two minimization objectives of cost and delay subject to satisfaction of reliability and flow constraints. (In the past, EAs have been extensively used in *single* objective optimization for various communication network related design problems - we give a brief survey of such work in the next section.) The remainder of the paper is organized as follows. In section 2, we present the related work done for communication network design problem. We describe, in section 3, a suitable model for the representation of a communication network and its implementation. Then, we present results in section 4 along with a comparison with the conventional methods. Finally, we draw conclusions in section 5.

2 Related Work

Since Network Design Optimization is an NP-hard problem, heuristic techniques have been used widely for such design. Heuristic methods that have been used include techniques, such as branch exchange, cut saturation etc. For example Jan et al. developed a branch and bound based technique to optimize network cost subject to a reliability constraint [18]. Ersoy and Panwar developed a technique for the design of interconnected LAN and MAN networks to optimize average network delay [19]. Clarke and Anandalingam used a heuristic to design minimal cost and reliable network [20]. However, these being heuristics, they do not ensure that the solutions obtained are optimal. Some of these heuristics evaluate trees and thus a large number of possible solutions are left unexplored.

Linear and Integer Programming has been used to a limited extent for network optimization since the number of equations varies exponentially with the number of nodes [21]. Also, greedy randomized search procedures [22] and other meta heuristics have been used for combinatorial optimization.

EAs have been extensively used in single objective optimization for many communication network related optimization problems. For example, Baran and Laufer [23] presented an Asynchronous Team Algorithms (A-Team) implementation, in a parallel heterogeneous asynchronous environment, to optimize the design of reliable communication networks given the set of nodes and possible links. The proposed Team combines parallel GAs, with different reliability calculation approaches in a network of personal computers. Abuali et al. assigned terminal nodes to concentrator sites to minimize costs while considering maximum capacity [24]. Ko et al. used GA for design of mesh networks but the optimization was limited to optimizing the single objective of cost while keeping minimum network delay as a constraint [25]. Elbaum and Sidi used GA to design a LAN with the single objective of minimizing network delay [26]. Kumar et al. used GA for the expansion of computer networks while optimizing the single objective of reliability [27]. White et al. used GA to design Ring Networks optimizing the single objective of network cost [28]. Dengiz et. al [29] presented a EA with specialized encoding, initialization, and local search operators to optimize the design of communication network topologies.

Most approaches attempted to optimize just one objective. For some approaches, the problem is broken down into a number of subproblems, solved in sequence using some heuristics thereby possibly leading to locally optimal design. Ravi et al. [8] and Deo et al. [6] presented approximation algorithm by optimizing one criterion subject to a budget on the other. Since then, many polynomial-time algorithm have been developed for several NP-hard optimization problems arising in network design. Different connectivity requirements such as spanning trees, Steiner trees, generalized Steiner forests, and 2-connected networks have been considered.

However, a practical multiobjective optimization approach should *simultaneously* optimize multiple objectives subject to satisfiability of multiple constraints. In this work, we present a framework using EAs that simultaneously optimize multiple objectives and produces a set of non-dominated *equivalent* solutions that lie on (near-) optimal Pareto- front.

3 Design and Implementation

Problem Definition: Topological design of WANs involves determining the layout of links between nodes given the mean/peak inter node traffic such that certain parameters of the network are optimized. In the solution developed, the total network cost and average delay on links is minimized simultaneously to obtain a Pareto front of optimal non-dominated solutions.

Design Parameters: For design, we use the following network parameters: the total number of nodes in the network N, the distance matrix D_{ij} which gives the physical distance between nodes i and j in kms, the traffic matrix T_{ij} which gives the expected peak network traffic between nodes i and j in packets per second, the number of types of network equipment slabs available K, and the number of types of link slabs available M along with the link cost per unit distance and link capacity.

Objective Functions: We use two objective functions - cost and delay - each of which is approximated by the following formulation:

1. **Cost:**
   $Cost = Costnodes + Costlinks + Costamps$
   where,
   $Costnodes = \sum_i C_i$; C_i = cost of the network equipment placed at node i
   $Costlinks = \sum_i \sum_j C_{ij}$; C_{ij} = cost of the link between node i and node j

   $Costamp = \frac{\sum_i \sum_j D_{ij} \times A}{L}$; L = maximum distance for which the signal is sustained without amplification, and A = cost of each amplifier unit.
2. **Average Delay:**

   $$AvgDelay = \frac{\sum_i \sum_j (Delay_{ij} \times LinkFlow_{ij})}{\sum_i \sum_j LinkFlow_{ij}}$$

   $LinkFlow_{ij} = \sum_k \sum_l Traffic_{kl}$ $\forall\, k, l$ nodes in the network such that the route from node k to node l includes the link (i, j). From queuing theory,

   $$Delay_{ij} = \frac{1}{Cap_{ij} - LinkFlow_{ij}}$$

   $Delay_{ij}$ is the link delay for packets flowing through link (i, j), and Cap_{ij} is the capacity of link (i, j). $LinkFlow_{ij}$ and $Delay_{ij}$ are 0 if there is no link between nodes i and j. $AvgDelay$ is ∞ if the network cannot handle the required traffic pattern with the existing capacities of the links and the routing policy adopted.

Constraints: Optimization of cost and delay functions are done subject to the following constraints:

1. **Flow Constraint:** Flow along a link (i, j) should not exceed the capacity of the link. Checking whether the total traffic along a link exceeds the capacity imposes this constraint. If it does, then the network is penalized.
2. **Reliability Constraint:** The network generated has to be reliable. The number of articulation points is a measure of the unreliability of the network. An articulation point of a graph is a vertex whose removal disconnects the graph. The number of articulation points is determined, and this constraint is imposed penalizing the network proportional to their number.

Routing Policy: To calculate the traffic through a particular link the routes between the nodes have to be known so that by superposition principle the total traffic on a link can be calculated. Routing is dynamic in real life and at any point the delays on the various links calculated from the traffic flowing through them gives the best route to be evaluated from the traffic matrix. For solving the design problem at least a rough static route has to be obtained. Dijsktra's shortest path algorithm is used for routing. The metric used for this purpose is the length of the link.

Encoding: In the encoding scheme chosen, every chromosome encodes a possible topology for interconnecting the given nodes; i.e., a chromosome represents a network, which is an individual in a set of potential solutions of the problem. This set of potential solutions constitutes a population. A constant length bit string representation was used to represent the chromosome. The chromosome consists of two portions; the first portion containing details of the network equipments at the nodes and the second portion consisting of details of the links. For instance, if there are T types of nodes, then $\lceil log_2 T \rceil$ bits are needed to encode a node. Thus the first portion of the chromosome consists of $\lceil log_2 T \rceil \times N$ bits. If a link is present between nodes 1 and 2 then the first bit position in the link portion is set to 1. Thus, the second portion of the chromosome consists of $\frac{N \times (N-1)}{2}$ bits. For example, we take 4 bits to encode up to 16 types of nodes. So, the first part of the chromosome contains $4 \times 4 = 16$ bits and the second part of the chromosome contains $\frac{4 \times (4-1)}{2} = 6$ bits.

The capacity of the link is then the first capacity value in the link slab that is greater than the minimum of the capacities of the NE s at the two node ends.

Initial Population: We use hybridization of EAs and conventional algorithm in generating the initial population. The following steps are used to generate the initial population. The network equipments (NE) at the nodes are randomly assigned and maintained in the chromosome. Assuming that the individual is fully connected, a minimal spanning tree is generated using Prim's algorithm. All co-tree links are then removed. A random number of links is then added from the co-tree set to the spanning tree. The number of links added is a random number in between one-third of the total number of links to half of the total number of links. This is done so that the initial population is not limited to spanning trees. This way we adopt a hybrid approach so that the time for exploitation and exploration of the search space is significantly reduced, and the number of lethals produced for large nets is minimized.

Fitness Evaluation: We use Pareto-rank based EA implementation. The Pareto rank [11] of each individual is equal to one more than the number of individuals dominating it in the multiobjective vector space. All the non-dominated individuals are assigned rank one. The values of the two objectives to be minimized (cost and average delay) are used to calculate the rank of the individual. Using the superposition principle the traffic on each individual node is calculated and hence the average delay for the network is calculated. Based on these two objectives the rank of the individual is calculated. In this work, we calculate fitness of an individual by $Fitness = \frac{1}{(Rank)^2}$.

Other Genetic Operators: We use Roulette wheel selection for selecting the parents. We divide chromosome in two parts for crossover. In the first part of the chromosome, initially the crossover point would lie at any position in the chromosome irrespective of the boundaries of the bits encoding. Node type values are not preserved to ensure maximum exploration. As the algorithm proceeds the probability of getting a crossover point within a node's NE boundary in the chromosome is constantly reduced so as to exploit the collected experience regarding optimal values of NE types so far. In this case only the existing NE types in the parents can be present in the children. In the link portion of the chromosome, since a single bit is used to code the presence or absence of the link, such considerations regarding tradeoff between exploration and exploitation do not arise. As a result, the crossover point is purely random. We use multi-point crossover; the number of crossover points depends on the problem-size. We use a simple bit-flipping mutation to further increase the exploration of the solution space.

Presence of Unconnected Components: As a result of the crossover and mutation operations, unconnected networks are generated as offspring. We do not completely eliminate the unconnected networks from further consideration. We maintain a pool of unconnected networks. This may give rise to fitter and connected components after further evolutions. This approach of maintaining unconnected, unfit individuals separately in the population is in accordance with the philosophy that unfit individuals can produce fit children.

Ensuring Convergence: For this we compute Intra-Island Rank-Histogram for each epoch of the genetic evolution and monitor the movement of Pareto-front. Since, this is a *hard* problem, it is likely that the problem may get trapped in local optima. To ensure a global (near-) optimal Pareto-front, we use a multi-island approach and monitors the Pareto-front using Inter-Island Rank histogram. The computation of Rank-histogram is analogous to that given in [16].

4 Results

We collected data of mass communication networks of different cities to carry out the simulation. We used the data which was used by the researchers in their previous work. We tested the algorithm for networks with up to 36 nodes, and convergence to an optimal Pareto front was observed. We conducted the experiments with many sets of random populations, and analyzed many sets of results. We also compared results with those obtained from other approaches namely exhaustive search and the Branch Exchange heuristic. In the following subsections, we include a few representative results.

Network of 10 Chinese Cities: The GA was run for the same problem as solved by Ko et al. [25]. In brief, the problem consisted of designing a packet switched mesh communication network among 10 major Chinese cities with realistic topology and traffic requirements. The design assumed a cost structure proportional to the distance among nodes and accounted for three different line rates: 6, 45 and 150 Mbps.

For a set of initial population of size 100, the solution space was found to improve very quickly up to the 40^{th} epoch. Then the improvement was marginal. We carried the evolutions up to the 100^{th} epoch. The rate of improvement was observed to be very slow; this was monitored by a rank-ratio histogram [16]. We include the initial population and the population at 60^{th} epoch in Fig. 1.

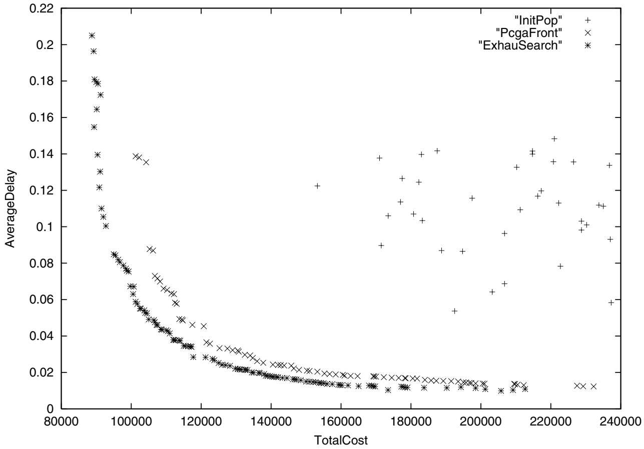

Fig. 1. 10 node network : (a) Initial population, (b)the converging Pareto-front obtained from EA and (c) the optimal Pareto-front obtained from exhaustive search.

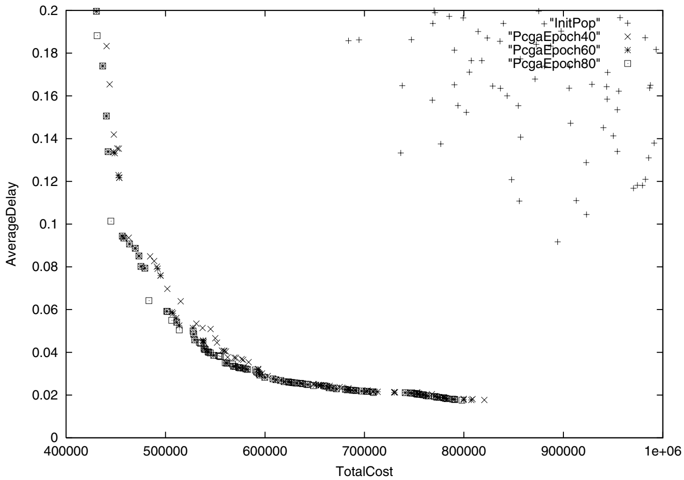

Fig. 2. A converging Pareto-front for a 21-node network. The population of size 100 is converging slowly to the Pareto-front during the later stages of evolution.

Network of 21 US Cities: Next, we tested our algorithm on a problem with larger number of nodes. This is a more complex than the earlier problem, so the improvement with epochs was slower.

Figure 2 shows the initial population and the non-dominated points obtained at epochs 40, 60 and 80. As seen from the plots, the movement of the Pareto Front is very-very marginal after the 40^{th} epochs. However, a few new solutions were being added to the Pareto-front with evolutions in low-cost and high-delay region. We observed that finding unformly distributed diverse solution in this non-linear region was a difficult task. However, we obtained diversity in this region by running EA for longer epochs. Alternately, this could be done by adopting the multi-island approach and by assessing convergence using inter-island rank histogram [16].

Network of 36 European Cities: Finally, we ran EA for a problem with 36 nodes. This is much more complex than the previous two problems. It took much more computational resources; we started with a population of 250 size and we could get nearly converged solution space at some 60^{th} epochs. This is shown in Fig. 3. The improvement was significant but with slower rate. The behavior of the population dynamics was quite analogous to the earlier results obtained with smaller and medium sized networks.

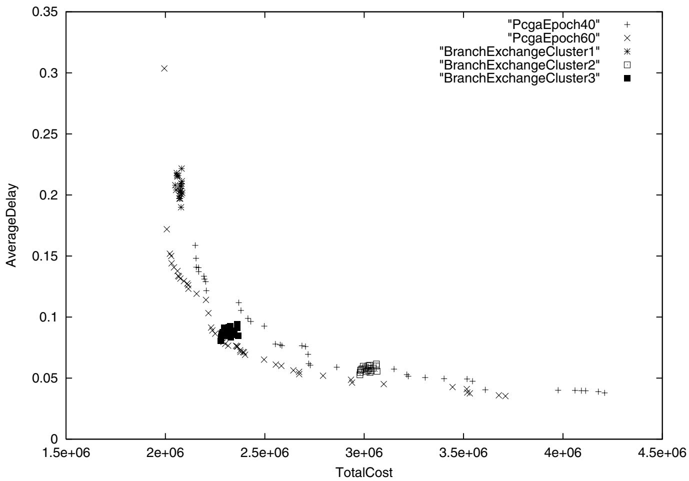

Fig. 3. A 36-node network: Comparison of solutions obtained by EA with the clusters obtained from Branch Exchange Heuristics.

Comparison: In order to show the relative merits of the EA approach we compared the results with those obtained by two of the commonly used conventional methods, namely, Exhaustive Search and Branch Exchange Heuristic.

An exhaustive search was done for all possible networks of size $N = 10$ nodes. All the possible networks were generated and evaluated, then all the non-dominated solutions in a given range were plotted against the results obtained with the Genetic Algorithm in Fig. 1. Since the problem is NP hard, the exhaustive search is of exponential complexity and is completely unfit for networks with more than 10 nodes. The complexity of the exhaustive search was found to be $O(2^{N^2})$. This is because there are $2^{{}^N C_2}$ graphs possible with N nodes. The deterministic solution is slightly superior to the results obtained by EA. This is expected because the deterministic Algorithm exhaustively searches all possible topologies. But it is also observed that the difference between the results obtained by Exhaustive Search and Genetic Algorithm is quite close. This gap is specific to a solution space which was obtained by a random sampling of the initial population; secondly this was not run to a *total* convergence. A multi-island approach as suggested in [16] is one of the possible solution to obtain a *superior* convergence. This is an area of further investigation.

The price paid for this marginal improvement obtained by Exhaustive Search over a single island Genetic Algorithm is the computation time involved. It was observed that for $N = 10$ node network the Exhaustive Search took more than 10 hours on a typical Intel Pentium P-IV, 1.7 MHz machine, whereas, the GA took a couple of minutes only. We could not compute the results for $N > 10$ nodes because of the exponential nature of the problem.

Another conventional method widely used for network optimization problems is Branch Exchange Heuristic. Many authors have used this heuristics to compare the results obtained by their algorithms. So we also compare the results obtained by EA with the clusters obtained by the Branch Exchange method. Here, we use an ϵ-constraint Branch Exchange to extend its use to multiobjective optimization. Different constraints have been put on any one objective function to obtain the solution in different regions of the Pareto-front. A few clusters are depicted in Fig. 3. As observed in Fig. 3 the results obtained by the Branch Exchange algorithm are comparable to a subset of the solutions obtained by EA but the diversity of the branch exchange is much less compared to that of EA. This is due to the fact that the branch exchange method considers only those network topologies that are spanning trees. However, such heuristics are unable to obtain most regions of the Pareto-front. This is a distinct advantage of EA in solving such *hard* problem.

5 Discussion and Conclusions

In this work, we demonstrated the solution of optimizing topologies of communication networks subject to their satisfying the twin objectives of minimum cost and delay along with two constraints. The solution to the network design problem is a set of optimal network topologies that are non-inferior with respect to each other. The multiple objectives to be optimized have not been combined into one and hence the general nature of the solution is maintained. These topologies are reliable in case of single link failures and it is guaranteed that the maximum packet load on any link will not exceed the link capacity. Thus the network is two edge connected and satisfies the constraints.

The algorithm has been test run on small as well as large networks. The initial population used in EA was taken from some hybridization of spanning tree and random topologies. The initial population and the final front were located far apart (Figs. 1 and 2). As a result much of the optimization was done by EA.

In most optimization problems like network design, it is crucial for the final solution-space to be diverse. As is observed from the results, EA achieves greater diversity in *polynomial time* as compared to other methods considered. A network designer having a range of network cost and packet delay in mind, can examine several optimal topologies simultaneously and choose one based on these requirements and other engineering considerations. The solutions obtained by traditional approaches do not show diversity. This is the primary advantage of using Pareto-rank based techniques to solve multiobjective optimization problems of such a hard nature.

Acknowledgements. This research is supported by Ministry of Human Resource Development(MHRD), Government of India project grant.

References

1. M. R. Garey and D. S. Johnson. *Computers and Interactability: A Guide to the Theory of NP-Completeness*, 1979. San Francisco, LA: Freeman.
2. D. Hochbaum (Ed.). *Approximation Algorithms for NP-Hard problems*, 1997. Boston, MA: PWS.
3. M. Gerla and L. Kleinrock. On the topological design of distributed computer networks. *IEEE Trans. Communications*, 25(1): 48–60, 1977.
4. V. P. Kompella, J. C. Pasquale, and G. C. Polyzos. Multicast routing for multimedia communication. *IEEE/ACM Trans. Networking*, 286–292, 1993.
5. M. Borah, R. M. Owens, and M. J. Irwin. An edge-based heuristic for Steiner routing. *IEEE Trans. Computer Aided Design of Integrated Circuits and Systems*, 13(12): 1563–1568, 1995.
6. N. Boldon, N. Deo, and N. Kumar. Minimum-weight degree-constrained spanning tree problem: Heuristics and implementation on an SIMD parallel machine. *Parallel Computing*, 22(3): 369–382, 1996.
7. M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt. Bicriteria network design problems. *J. Algorithms*, 28(1): 142–171, 1998.
8. R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt. Approximation algorithms for degree-constrained minimum-cost network design problems. *Algorithmica*, 31(1): 58–78, 2001.
9. C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2002. Boston, MA: Kluwer.
10. K. Deb. *Multiobjective Optimization Using Evolutionary Algorithms*, 2001. Chichester, UK: Wiley.
11. C. M. Fonseca and P. J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms – Part I: a unified formulation. *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, 28(1): 26–37, 1998. 26–37.
12. K. Deb et al. A fast non-dominated sorting genetic algorithm for multiobjective optimization: NSGA-II. *Parallel Problem Solving from Nature*, PPSN-VI: 849–858, 2000.
13. E. Zitzler, M. Laumanns and and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. *EUROGEN* 2001.
14. Knowles, J. D. and Corne, D. W. Approximating. *Evolutionary Computation*, 8(2): 149–172, 2000.

15. M. Laumanns, L. Thiele, K. Deo and E. Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3): 263–182, 2002.
16. R. Kumar and P. I. Rockett. Improved sampling of the Pareto-front in multiobjective genetic optimizations by steady-state evolution : a Pareto converging genetic algorithm. *Evolutionary Computation*, 10(3): 283–314, 2002.
17. R. C. Purshouse and P. J. Fleming. Elitism, sharing and ranking choices in evolutionary multi-criterion optimization. Research Report No. 815, Dept. Automatic Control & Systems Engineering, University of Sheffield, Jan. 2002.
18. R. H. Jan, F. J. Hwang, and S. T. Cheng. Topological optimization of a communication network subject to a reliability constraint. *IEEE Trans. Reliability*, 42(1): 63–69, 1993.
19. C. Ersoy and S. S. Panwar. Topological design of interconnected LAN/MAN Networks. *IEEE J. Select. Areas Communication*, 11(8): 1172–1182, 1993.
20. L. W. Clarke and G. Anandalingam. An integrated system for designing minimum cost survivable telecommunication networks. *IEEE. Trans. Systems, Man and Cybernetics- Part A*, 26(6): 856–862, 1996.
21. A. Atamturk and D. Rajan. Survivable network design: simultaneous routing of flows and slacks. Research Report, IEOR, University of California at Berkeley.
22. T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 1995.
23. B. Baran and F. Laufer. Topological optimization of reliable networks using A-Teams. National Computer Center, National University of Asuncion, University Campus of San Lorenzo – Paraguay.
24. F. N. Abuali, D. A. Schnoenefeld, and R. L. Wainwright. Designing telecommunication networks using genetic algorithms and probabilistic minimum spanning Trees. In *Proc. 1994 ACM Symp. Applied Computing*, pp. 242–246, 1994.
25. K. T. Ko, K. S. Tang, C.Y. Chan, K. F. Man and S. Kwong. Using genetic algorithms to design mesh networks. *IEEE Computer*, 6–58, 1997.
26. R. Elbaum and M. Sidi. Topological design of local-area networks using genetic algorithms. *IEEE/ACM Trans. Networking*, 4(5): 766–777, 1996.
27. A. Kumar, R. M. Pathak, and Y.P. Gupta. Genetic-algorithm based reliability optimization for computer network expansion. *IEEE Trans. Reliability*, 44(1): 63–72, 1995.
28. A. R. P White, J. W. Mann, and G. D. Smith. Genetic algorithms and network ring design. *Annals of Operational Research*, 86: 347–371, 1999.
29. B. Dengiz, F. Altiparmak, and A. E. Smith. Local search genetic algorithm for optimal design of reliable networks. *IEEE Trans. Evolutionary Computation*, 1(3): 179–188, 1997.