Multiobjective Network Design for Realistic Traffic Models

Nilanjan Banerjee
Department of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
nilanb@cs.umass.edu

Rajeev Kumar*
Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur
Kharagpur, WB 721 302, India
rkumar@cse.iitkgp.ernet.in

ABSTRACT

Network topology design problems find application in several real life scenarios. However, most designs in the past either optimize for a single criterion like delay or assume simplistic traffic models like Poisson. Such assumptions make the solutions inapplicable in the practical world.

In this paper, we formulate and solve a multiobjective network topology design problem for a realistic Internet traffic model which is assumed to be self similar. We optimize for the average packet delivery delay and network layout cost to construct realistic network topologies. We present a multiobjective evolutionary algorithm (MOEA) to obtain the diverse near-optimal network topologies. For fair comparison, we design a multiobjective deterministic heuristic based on branch exchange – we call the heuristic Pareto Branch Exchange (PBE). We empirically show that the MOEA used performs well for real networks of various sizes, and generated topologies are quite different with significantly larger delays for the self similar traffic model.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods and Search—*Heuristic Methods*; G.1.6 [**Optimization**]: Stochastic programming; J.6 [**Computer-Aided Engineering**]: Computer-Aided Design.

General Terms

Algorithm, Design, Experimentation.

Keywords

Optimization methods, combinatorial optimization, genetic algorithm, multiobjective optimization, heuristics, Pareto front, network topology design, self-similar traffic.

*Author acknowledges receiving financial support from a research project grant funded by Ministry of Human Resource Development (MHRD), Government of India.

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

1. INTRODUCTION

Single objective network design problems are well studied and many heuristics exist for obtaining exact/approximate solutions in polynomial-time [11, 13]. However, in real-life applications, network design problems generally require simultaneous optimization of multiple objectives, subject to satisfaction of constraints. Moreover, network traffic in real life applications use traffic models which are far from the widely used Poisson model.

For example, consider a topology design problem of the scale of the Internet. Such a topology design should consider simultaneous optimization of network cost, average delay, and number of packets delivered subject to reliability, bandwidth and/or flow-constraints. Moreover as shown in [12, 24] , the traffic model considered should be self similar in nature.

Thus, in this work, we formulate a general bi-criteria bi-constrained communication network topology design problem, and solve this real-world application (RWA) using a general multi-objective evolutionary algorithm. We consider average network delay and network cost as the two optimization objectives subject to satisfaction of reliability and flow constraints. For delay, we consider self-similar traffic models [12, 24] and compare the topologies with those obtained through the Poisson model.

The formulation, analysis and evaluation of the multi-objective evolutionary algorithm for the realistic network topology design presents three novel research contributions.

- Through an extensive set of simulations on real world data, this paper presents an analysis of difference in networks produced by realistic self-similar model to Poisson traffic model.
- It presents a general, computationally inexpensive tool for solving NP-Hard multiobjective optimization problems. The scheme allows a simple and natural selection process to maintain diversity and assessing convergence.
- The work also presents a deterministic multiobjective heuristic (Pareto Branch Exchange heuristic) to solve the above problem.

The rest of the paper is organized as follows. In Section 2 we present a brief review of the communication topology design and the issues and challenges in solving multiobjective real-world applications using EAs. We describe, in Section 3, the formalization for communication network topology optimization problem. In Section 4 we describe the evolutionary algorithm used for the topology design. Section 5 presents exhaustive search and Pareto Branch exchange heuristic. The empirical results are presented in Section 6. Finally, conclusions are drawn in Section 7.

2. COMMUNICATION NETWORK DESIGN – A REVIEW

There are several research monograms in the literature which formulate heuristics and meta-heuristics for the efficient design of *general graph* based network topologies. The primary goal is to design low cost reliable networks. Since most non-trivial network topology design problems are NP-hard in nature some heuristics have been used such as branch exchange, and cut saturation — see Frank and Chou [10] for a tutorial on the subject. Subsequently, many studies have been done for optimizing average network delay and designing minimal cost reliable networks, e.g., [3, 14].

Some of these heuristics evaluate trees and thus a large number of possible solutions are left unexplored. Multi-criteria spanning tree problem is the most studied special case of a multicriteria network design problem. For example, Deo et al. [7], Raidl and Julstorm [22], and Ravi et al. [23] present heuristics/approximation algorithms for solving degree/diameter-constrained spanning trees. Zhou and Gen [25], Knowles and Corne [15] and Kumar et al. [20] use multiobjective evolutionary algorithm (MOEA) to obtain multiple solutions simultaneously for bi-objective spanning tree problems. However, results on *general* graphs are sparse.

Linear and integer programming have been used to a limited extent for network optimization since the number of variables grows exponentially with the number of nodes, e.g., [1, 2]. There are several randomized search techniques that were used for design of network topologies. Simulated annealing and tabu search have been used for the design of network topologies, however, the network architecture is confined to spanning trees. Greedy randomized adaptive search procedures (GRASP), a multi-start meta heuristic, has also been used in network design [8].

In recent years, a lot of interest has been generated among researchers in using EA for solving communication network design problems. Ko et al. used EA for the design of mesh networks but the optimization was limited to the single objective of cost with minimum network delay as a constraint [16].

However, a practical multiobjective optimization approach should *simultaneously* optimize multiple objectives subject to satisfiability of multiple constraints and output a set of solutions forming a Pareto-front. We presented few initial results of network topology design for Poisson traffic model in [17, 18]. 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. Moreover, we use Poisson and self-similar traffic models to calculate the average delay in the network showing the applicability of EA to network topology design using realistic queuing models.

In the multiobjective scenario, EAs often effectively find a set of mutually competitive solutions without needing much problem-specific information. However, achieving proper diversity in the solution-set while approaching convergence is a challenge in MOO, especially for problems whose solution front is not known *a priori*. Many techniques and operators have been proposed to achieve diversity. A common metric for convergence is the distance metric, which finds distance of the obtained solution front from the true Pareto front; this is trivial for known problems. Such a metric is based on a reference front. In real-world search problems, location of the actual Pareto-front is unknown. A commonly practiced approach to determine the reference front for unknown problems is to extract the reference front from the best solutions obtained so far, and the reference is incrementally updated with every generation using an iterative refinement based scheme.

3. PROBLEM FORMULATION

Topological design of WANs involve determining the layout of links between nodes – given the mean/peak inter node traffic. In the solution developed, the total network cost and average delay on links are minimized simultaneously. We formally define the network topology problem below.

3.1 Design Parameters

For the design, we use the following network parameters:

- the total number of nodes in the network N,
- a distance matrix D_{ij} which provides the physical distance between every pair of nodes i and j,
- a traffic matrix T_{ij} which provides the expected peak network traffic between every pair of nodes i and j,
- the number of types of network equipment (NE) slabs available K, and the number of types of link slabs available M along with the link cost per unit distance and the link capacity, and
- the traffic delay models used—for example, *Poisson* and *Self-similar*

3.2 Objective Functions

We use two objective functions—network cost and end-to-end message delivery delay—each of which is *approximated* by the following formulation.

3.2.1 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, and
- $CostAmps = A \times \sum_i \sum_j \lfloor D_{ij}/L \rfloor$; A = cost of each amplifier unit, and L = maximum distance for which the signal is sustained without amplification.

The cost function considers the cost of laying down the nodes and the links. Further, it also considers the cost of laying down amplification units on the links for proper attenuation of the signals. Note that amplifier units form an important part of real networks.

3.2.2 Average Delay

We model the traffic between nodes as a *Poisson* and *Self-similar* process. Although, *Poisson* process has been vastly used in the past to model traffic in conventional networks, recently it has been shown that Internet traffic exhibits self-similar bursts [12, 24]. We provide a mathematical formulation for the two traffic models below.

Queuing delay for Poisson Traffic. The delay in a network for the queuing model is mainly due to the queuing of packets in intermediate nodes. The delay formalization can be stated as follows:

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

where $LinkFlow_{ij} = \sum_k \sum_l Traffic_{kl}, \forall\ k, l$ nodes in the network such that the route from node k to node l includes the link $(i,\ j)$. From queuing theory, the queuing delay $Delay_{ij}$ using standard M/M/1 (Poisson) queuing model is given by,

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

where 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 links and routing policy adopted.

Queuing Delay for Self-Similar Traffic. For bursty traffic, there is no natural length for a burst; traffic bursts appear on a wide range of time scales and the traffic is self-similar over a large time-scale. It has been shown by many researchers that Internet traffic is better modeled using self-similar processes, e.g., [21] . There are many models proposed for such traffic. One of them is the multiplexed ON/OFF traffic pattern with Long-Range Dependent (LRD) properties. Hans-Peter Schwefel, in his Ph.D thesis [24], developed techniques for analysing queuing models for such traffic. We use, in this work, the delay formulation proposed by Schwefel. The average delay, under certain approximation [24], is given by,

$$mPD(MBS) \sim c_{mPD} MBS^{2-\beta}$$

where,

- Maximum Burst Size (MBS) is defined as the approximate Power Tail (PT) Range of the distribution of the number of packets. The bursts with more than MBS packets only happen with very small probability.

- The tail constant c_{mPD} can be obtained as follows:

  $$\overline{c_{mPD}} \approx \frac{1}{\omega}.\frac{\rho}{1-\rho}.\frac{i_\Delta}{i_0^{2-\beta}}.\overline{n_p}^{\beta-1} b^2 (1-b)^{i_0-1}.\frac{[c_{PT}^{(1)}(\alpha)]^{i_0}}{(2-\beta)(\alpha-1)^{i_0}}$$

  where $\overline{n_p}$ is the mean number of packets per burst and $c_{PT}^{(1)}(\alpha)$ is the tail constant of normalized PT distribution.

- ω is defined as the average cell rate. For N sources, $\omega = N.\kappa + \omega_0$, where κ is the average traffic rate of the ON and OFF times together and ω_0 is the rate of background Poisson traffic.

- The *burstiness parameter* b is defined as $1 - \frac{\kappa}{\omega_p}$ where ω_p is the peak rate at which traffic is generated during the ON period of a single source.

- β is the tail exponent and is given by $\alpha_0(\alpha - 1) + 1$ where α is the power exponent.

- If ν is the service-rate, the utilization is defined as $\rho = \frac{\omega}{\nu}$.

- For $\omega_0 = 0$ the blow-up region i_0 is given by

  $$i_0 = \left\lceil N.\frac{1-\rho}{\rho}.\frac{1-b}{b} \right\rceil$$

- The other parameter i_Δ is defined as follows

  $$i_\Delta = i_0 - N.\frac{1-\rho}{\rho}.\frac{1-b}{b}$$

  which has a range $0 \leq i_\Delta < 1$.

Minimization of cost and delay is 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.

2. *Reliability Constraint.* The network generated has to be reliable. The number of *articulation points* is a measure of how unreliable the network is. An articulation point of a graph is a vertex whose removal disconnects the network graph.

Dijkstra's shortest path algorithm is used to determine the path to route traffic from a source to a destination [5]. The weights of the links in the algorithm is taken as proportional to the link length.

4. MOEA FOR TOPOLOGY DESIGN

The network topology problem formulated above is a complex discrete non-linear NP-Hard problem. It is difficult to define bounds on the solution space for such a problem – thus difficult to define convergence criteria for a heuristic. Moreover, the solution space can be potentially exponential in size – computationally expensive to search.

There are many MOEAs and their implementations. We chose the Pareto Converging Genetic Algorithm (PCGA) [19], which is a steady-state algorithm that has been shown to perform well across a wide variety of non-linear optimisation problems.

The PCGA algorithm used in this work is a steady-state algorithm and can be seen as an example of $(\mu + 2)$ – Evolutionary Strategy (ES) in terms of its selection mechanism [4, 6]. In this algorithm, individuals are compared against the total population set according to a the Pareto-ranking scheme [9] and the population is selectively moved towards convergence by discarding the lowest ranked individuals in each evolution. The process is iterated until a convergence criterion based on Intra-island rank-ratio and

Algorithm 1 Pareto Converging GA

Input: N - size of initial population and GA parameters
Output: a set of (near-) optimal solutions
Algorithm:
Generate an initial population of size N
Compute individual's objective vector
Pareto-rank the population and generate rank-histogram
while Intra-island rank-ratio histogram does not satisfy stopping criteria **do**
    Select two parents using roulette wheel selection scheme
    Perform crossover and mutation to generate two offspring
    Compute objective vectors of offspring
    Pareto-rank the population including offspring
    Remove the two least fit individuals (with tie resolution) to keep the size N
    Generate rank-ratio histogram
end while
One while-loop for Inter-island rank-histogram satisfying stopping criterion
Output set of solutions

Inter-island rank histogram is achieved [19]. The pseudo code of the baseline PCGA is included in Algorithm 1.

The initial population is generated through a mix of randomization and a deterministic algorithm. The network equipments (NE) at the nodes are randomly assigned and maintained in the chromosome. Assuming that the network is fully connected, a minimal spanning tree is generated using Prim's algorithm [5]. All co-tree links are then removed. A random number of links is then added from the co-tree set to the spanning tree. Through the links of the spanning tree topology the algorithm takes leverage of the fact that all optimal topology would be supersets of spanning tree. However, randomization is introduced in the initial population to eliminate the possibilities of the algorithm being trapped within a subset of spanning tree topologies – a local optima.

Each chromosome encodes a possible network topology. A set of such chromosomes forms the population. Each chromosome consists of a constant length bit string. The structure of the chromosome is illustrated in Figure 1. The chromosome comprises of two portions — the first portion contains details of the network equipments at each of the nodes and the second portion the details of the links. The size of chromosome depends on the types of network nodes, and the number of nodes in the network. If there are T types of nodes, then $\lceil log_2 T \rceil$ bits are used to encode a node. Thus, the first portion of the chromosome will have $\lceil log_2 T \rceil \times N$ bits. If a link is present between nodes 1 and 2 then the first bit position in the link portion is set to 1. The second portion of the chromosome, therefore, will have $\frac{N\times(N-1)}{2}$ bits. For example, consider the network topology in Figure 1, 2 bits are used to encode 4 types of nodes. The first part, hence contains 12 bits and the second part of the chromosome contains 15 bits.

Fitness of a chromosome is evaluated based on principle of Pareto ranking. Pareto-rank [9] of each individual is equal to one more than the number of individuals dominating it. All non-dominated individuals are assigned rank one. Network cost and average delay is used to evaluate the rank of an individual chromosome using the principle of Pareto dominance. The fitness of an individual is given by $Fitness = \frac{1}{(Rank)^2}$. We verified through experimentation that an inverse quadratic relation produces the best results.

Parent chromosomes for crossover is selected using the roulette wheel process. Since, the chromosome consists of two distinct parts, separate crossover is used for the two parts. For the first half of the chromosome, the crossover point can lie at any position of the chromosome irrespective of the boundaries of the bit encoding Since, the chromosome consists of two distinct parts, separate crossover is used for the two parts. For the first half of the chromosome, the crossover point can lie at any position of the chromosome irrespective of the boundaries of the bit encoding. During the first few iterations, the node type values are not preserved to ensure maximum exploration of the solution space. As the algorithm proceeds, the probability of getting a crossover point within a node's equipment-type boundary in the chromosome is reduced to exploit the collected experience regarding optimal values of NE types so far. In this case only the existing equipment-types in the parents can be present in the children.

In the link portion of the chromosome, the crossover point is chosen uniformly at random. We use multi-point crossover; the number of crossover points depends on the problem-size. In addition to crossover, a chromosome is mutated using random uniform mutation of every bit.

As a result of the crossover and mutation, unconnected networks can be generated as offsprings. We maintain a pool of unconnected networks. We argue that such a pool might have useful genetic material and consequently may give rise to optimal networks.

Convergence is assessed using *Intra*-Island rank-histogram for each epoch of the genetic evolution. However, it is likely that the solution may get trapped in a local optima or a plateau of sub-optimal solutions. To ensure that the algorithms does not get trapped within a plain of sub-optimal solutions, we use a multi-island approach that monitors the Pareto-front using *Inter*-Island rank histogram – for details see [19].

5. DETERMINISTIC PARETO HEURISTIC

Heuristic approaches used in the past for network topology design produce spanning/Steiner tree topologies which effectively optimize only one criterion — the average delay or the cost of the network. For example, Deo et al. [7], and Ravi et al. [23] presented approximation algorithms for optimizing one criterion that satisfy the degree/diameter constraint. In this section, we present a heuristic search algorithm which is multiobjective and tries to simultaneously minimizes average cost and the delay of the network — this heuristic is adapted from Branch Exchange algorithm [10], we call it Pareto Branch Exchange (PBE) heuristic.

Before presenting the PBE, we present the exhaustive search approach to obtain the Pareto-front. The solutions obtained is used to compare the those obtained from EA and PBE results obtained for smaller networks.

5.1 Pareto Exhaustive Search

In this method all possible networks are evaluated and Pareto ranked to obtain the set of Pareto-optimal networks. The complexity of the exhaustive search turns out to be

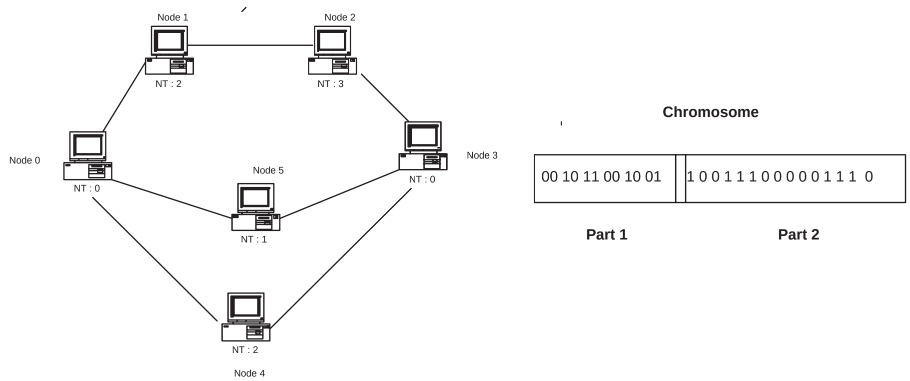

Figure 1: Encoding scheme for the chromosome. The left sub figure illustrates the network and the type of network equipment at each node. The right sub figure shows the chromosome structure.

$O(2^{N^2})$, where N is the number of nodes in the network. Thus, we could apply this method for networks of smaller sizes only.

5.2 Pareto Branch Exchange Heuristic

We generalize the conventional single objective branch exchange to the multiobjective case. The algorithm should serve dual purpose; one, it should simultaneously optimize the two objectives cost and delay, second, it should satisfy the flow and reliability constraints. Therefore, the algorithm has two phases. In the first phase, the algorithm constructs all possible spanning trees which are (near-) optimal with respect to the objective functions. However, such a topology may not assure bi-connectivity and satisfy flow constraints. Hence, in the second phase, edges are added to assure bi-connectivity.

PBE Heuristic : Phase 1

- It is assumed that the given network is a complete graph G. The traffic is routed between the required sources and destinations using the shortest path algorithm.
- *For* all the edges in the graph the cost and the delay is evaluated. Thus, we have a duple of cost and delay for each possible edge of the network.

  For all nodes N_i in the network. {

  - Construct a Spanning Tree S_i with N_i as the root;
  - Initialize an archive A_i which stores the non-dominated topologies.
  - The center of the tree is found and the paths from the leaves of the tree to the center is enumerated as $P_1, P_2, ..., P_k$;
  - *For* each path P_i {
    - ∗ The leaf node L is considered;
    - ∗ Its nearest neighbor N_l lying on a different path (say P_j) is taken and an edge is added between L and N_l , thus creating a cycle in the topology;
    - ∗ All the edges in the cycle and its associated duple (C, D) is considered for all the edges; ($\star$ C refers to the sum of the cost of the nodes connected by the edge, the edge cost and the amplification cost while the D refers to the queuing delay corresponding to the edge. $\star$)
    - ∗ The edges are ranked using the Pareto ranking scheme and the highest ranked edge is removed creating a new tree topology; ($\star$ If more than one edge have the same rank, the edge to be removed is selected at random. $\star$)
    - ∗ *For* the new topology the duple (C, D) is calculated;
    - ∗ *If* it dominates the previous topology, the topology is replaced and the new topology is stored in archive A_i;
    - ∗ *Else If* it is dominated by the previous topology, no change is made to the archive A_i;
    - ∗ *Else If* both the topologies are incomparable, both the topologies are stored in A_i;

    }

  }

- The archive set $A = \{A_1, A_2, ..., A_N\}$ is sorted by the Pareto ranks and the non-dominated topologies are taken as the solution.

End_of_Phase 1.

PBE Heuristic : Phase 2

- The non dominating spanning tree topologies from Phase 1 are taken as input.
- An archive R is initialized which stores the non-dominated reliable topologies.
- Assume that the given network as a complete graph G. The traffic is routed between the required source and destinations using the shortest path algorithm as in phase 1.
- *For* all the edges in the topology the duple (C, D) is evaluated for the graph G.
- All the edges thus formed are sorted according to their decreasing rank value and increasing fitness value in an archive G_m. Fitness is defined as $\frac{1}{Rank}$.

For all the spanning tree topologies T_i in A. {

– *EDGES_Label :* Consider the edges (say set S_i) which are not present in the spanning tree T_i

– Add edges in the topology according to the descending order of the sorted pool G_m to the spanning tree.

– Reroute the traffic checking the flow constraint and evaluate the duple (C, D) for the topology. *If* topology dominates the previous topology or is incomparable to it goto TEST_Label , *Else* goto EDGES_Label.

– *TEST_Label :* Test for reliability. *If* the topology is biconnected, then add topology to R . *Else* goto EDGES_Label.

}

- All topologies in R, evaluate duple (C, D) and do a Pareto ranking on all the topologies. The dominated topologies are removed from the archive R and the remaining are kept in R. R forms the final set of solutions.

End_of_Phase 2.

The PBE is a deterministic multiobjective heuristic and no approximation factor has been proven for the heuristic.

6. RESULTS & DISCUSSION

We carried-out an extensive set of simulations on real data set comprised of communication network data from (i) Network of 10 Chinese cities, (ii) Network of 21 U.S. cities, and (iii) Network of 36 European cities [16]. These data sets consisted of average communication traffic, and the average distance between nodes in the network. All simulations were run on an Intel Pentium P-IV, 1.7 MHz machine.

The input parameters to the EA – for example, the population size, the crossover and mutation probability – were tuned through a large number of runs. We used a small population of 100 individuals for the 10 and 21 node networks while a population of 250 chromosomes were used for the 36 node network. A larger population was chosen for the larger network to facilitate quicker search of the larger solution space. Each bit in the chromosome was mutated with a uniform probability of 0.01 for the 10 node network and with a probability of 0.02 for the larger networks. The mutation probabilities were chosen such that, on expectation, at least one bit gets mutated for smaller networks. However, for larger networks, the algorithm might possibly get trapped in a local optimal plateau surrounded by non-dominating solutions. Hence, a larger number of mutations are warranted to obtain the optimal solutions. The crossover probability is set to in the range of 0.7 to 1.0 to facilitate faster exploration of the search space.

For the 10 node network, with a set of initial population of size 100, good solutions evolved quickly within the first 100 epochs, and then the improvement was marginal for both the delay models. The rate of improvement was observed, on the Intra-island rank histogram, to be very slow. Figure 2 depicts the obtained Pareto-front for the Poisson delay models. The stopping criterion for the EA was judged

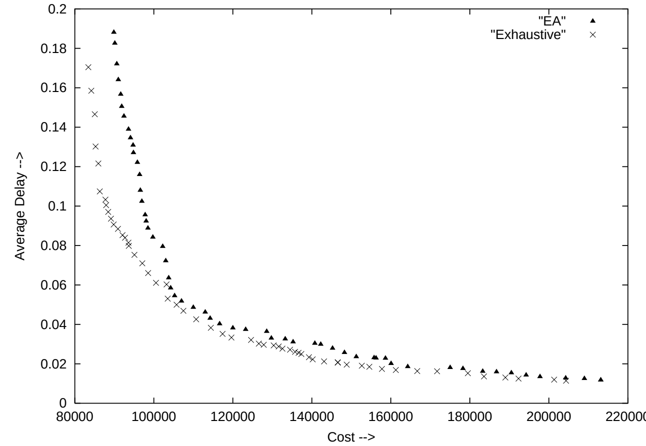

Figure 2: Comparison of results from exhaustive search (*optimal*) and EA for 10 node problem using *Poisson* traffic.

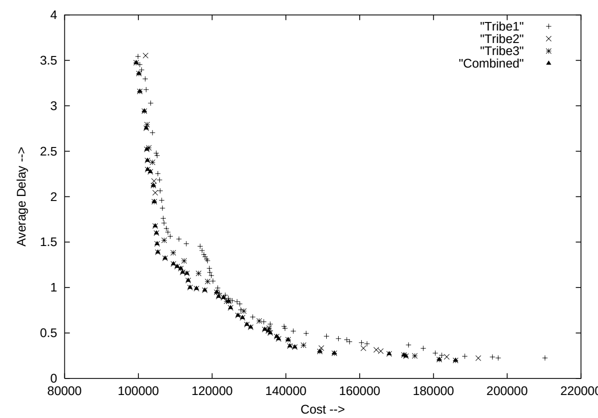

Figure 3: Solution sets (three tribes and a merged one) from EA using *self similar* traffic for 10 node problem.

using the intra-island rank ratio histogram technique. The simulation was run to its near-convergence for each of the tribes; it took epochs in the range of 600 – 800 runs for each of the tribes. We conducted experiments with multiple tribes. The number of tribes to be merged was based on the tail of Inter-island rank histogram. The merged solutions for Poisson and Self-similar delay models are included in Figures 2 and 3, respectively. It can be seen from the plots in obtained in Figure 3 that the combined curve is superior to the individual tribes as this contained genetic material from multiple-tribes. The topologies obtained for self similar traffic model exhibit much larger delay than those obtained by Poisson traffic.

For the small network, we also ran the Pareto exhaustive search to obtain the Pareto-optimal front. The front obtained from the exhaustive search is depicted in Figure 2 along with the combined-front obtained by merging the tribes using EA. The exhaustive search is superior to EA. However, the EA took approximately 10 minutes to obtain the

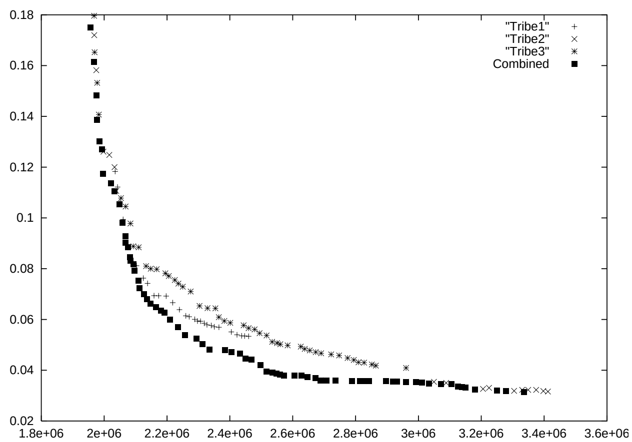

Figure 4: Solution sets (three tribes and a merged one) from EA using *Poisson* traffic for 36 node problem.

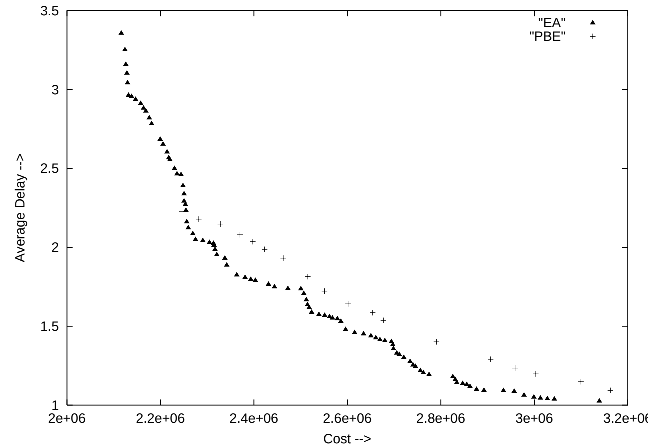

Figure 5: Comparison of the fronts of EA with the PBE for the 36 Node network for *self similar* traffic.

front while exhaustive technique needed around 40-hours. A marked difference in the two obtained solutions is that EA was not able to find good solutions in the low cost region which is a considerably steeper region.

We carried out experiments with the 36 node European city network. The experiment tests the scalability of the proposed heuristics. For each of the tribes, it took much more computational resources to obtain the near optimal front. The experiment started with a population of 250 and we could get nearly converged solution space approximately with 900^{th} epochs —Figure 4 depicts the solution fronts obtained for the Poisson model for three tribes. In the combined solution from all the three tribes in Figure 4, the high cost region is an outcome of the genetic material from tribe 2 while tribe 3 contributes significantly to the obtained topologies in the low cost region.

The behavior of the population dynamics was quite similar in all the cases considered above. However, we could not employ exhaustive search for larger networks as it is practically infeasible to run the exhaustive search algorithm for such networks. Therefore, we compared the solutions with those of the PBE heuristic. Results for the self-similar model are included in Figure 5. For both the traffic models, PBE results were inferior to those of EA results.

6.1 Characteristics of the Pareto front

For larger networks, the solution-fronts can be partitioned into three distinct regions – (i) Low Cost High Delay Region (LCHD), (ii) Medium Cost Medium Delay Region (MCMD), and (iii) High Cost Low Delay Region (HCLD). Solutions to the Low Cost High Delay (LCHD) Regions are easier to obtain. This is because they correspond to the spanning tree topologies. Most deterministic algorithms easily obtain this part of the front. However, obtaining the Medium Cost Medium Delay (MCMD) and High Cost Low Delay (HCLD) regions requires exploration of the larger solution space. It is clear from the solutions obtained, that the EA is able to efficiently explore the entire solution space and potentially obtain most of the Pareto-optimal solutions.

Although deterministic algorithms fail to obtain the entire solution front, yet the PBE obtains a good approximation of the solution front. This is because, the decisions to make a deterministic step forward in the algorithm is made on the basis of Pareto ranking. Therefore, the solutions are not limited to the LCHD regions but are spread across the (near-)optimal front.

The solutions obtained from the Poisson and Self-similar delay models are also shown to be different. For the same network cost, the delay introduced by the Self Similar traffic model is an order of magnitude larger than the Poisson delay model. Experimentation through different delay costs shows that the EA is able to obtain (near-) optimal solutions even for larger values of delay. This implies that the EA solution can be applied to optimization problems with a wide range of delay cost values.

Most deterministic heuristics which consider spanning trees only optimize for the network cost and get trapped in regions of high average delay and low network costs. Using $\epsilon-$constraint methods, or using multi-start deterministic heuristics do not help, since they are directed towards obtaining solutions in close vicinity of the original solutions. However, MOEA through use of random hops in the solution space is likely to explore the solutions space faster and more completely.

In complex non-linear optimization problems, like the network topology design problem the exploration process can get trapped in: (1) a sub-optimal solution, and/or (2) a plateau of local optimal solutions surrounded by sub-optimal solutions. Though a deterministic algorithm can possibly find its way from a sub-optimal solution to a an optimal solution, it is extremely difficult for many deterministic solution to detect that it is trapped in a plateau of locally optimal solution, surrounded by suboptimal data points. However, EA, can use global mutation (i.e., flipping every bit of the chromosome with a random uniform probability) or uniform crossover, to jump from a plateau of locally optimal solutions to a global optimal solution. Another important problem faced in such hard non-linear multiojective optimization is the diversity of the solutions obtained. Different starts of the EA can lead to multiple tribes of solutions on the Pareto front. However, the multi-start multi-tribe approach ensures that the genetic material from different tribes can be used to form the entire Pareto-optimal front.

7. CONCLUSIONS

In this work, a set of deterministic and stochastic/ randomized (EA) heuristics were presented for solving the multiobjective network topology design problem for realistic traffic models. We argue that combining different objectives into one objective and using a single objective optimization method to solve such problems is not very useful in this scenario. Therefore, there is a need to use tools which preserve the general nature of the problem and solve it using no a priori knowledge of the solution space. We show in the paper, that the EA heuristic generally provides better solutions than its deterministic counterparts. The topologies generated are reliable in case of single link failure 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 flow constraint. Topologies generated for self similar traffic have much higher delays than those of Poisson traffic. Therefore, through a mix of efficient random search and initial hybridization, EAs can be used as a general tool to solving such hard multiobjective optimization problems.

8. REFERENCES

[1] J. Arabas and S. Kozdrowski. Applying an evolutionary algorithm to telecommunication network design. *IEEE Trans. Evolutionary Computation*, 5(4), 309 - 322 2001.

[2] A. Atamturk and D. Rajan. Survivable network design: simultaneous routing of flows and slacks. Research Report, IEOR, University of California at Berkeley.

[3] 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.

[4] C. A. C. Coello, D. A. V. Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multiojective Problems*. Boston, MA: Kluwer, 2002.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2001.

[6] K. Deb. *Multiobjective Optimization Using Evolutionary Algorithms*. Chichester, UK: Wiley, 2001.

[7] N. Deo and P. Micikevicius. Comparison of prüfer-like codes for labeled trees. In *Proc. 32nd South-Eastern Int. Conf. Combinatorics, Graph Theory and Computing*, 2001.

[8] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 1995.

[9] C. M. Fonseca and P. J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part I: a unified formulation. *IEEE Trans. Systems, Man and Cybernetics-Part A: Systems and Humans*, 28(1):26 – 37, 1998.

[10] H. Frank and W. Chou. Topological optimization of computer networks. *Proc. IEEE*, 60(11):1395 – 1397, 1972.

[11] M. R. Garey and D. S. Johnson. *Computers and Interactability: A Guide to the Theory of NP-Completeness*. San Francisco, LA: Freeman, 1979.

[12] S. Ghose, R. Kumar, N. Banerjee, and R. Datta. Multihop virtual topology design in WDM optical networks for self-similar traffic. *Photonic Network Communications*, 10(2):199 – 214, Sep. 2005.

[13] D. S. Hochbaum. *Approximation Algorithms for NP-Hard Problems*. Boston, MA: PWS, 1997.

[14] 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.

[15] J. D. Knowles and D. W. Corne. A comparison of encodings and algorithms for multiobjective minimum spanning tree problems. In *Proc. 2001 Congress on Evolutionary Computation (CEC-01)*, volume 1, pages 544 – 551, May 27-30 2001.

[16] K. T. Ko, K. S. Tang, C. Chan, K. F. Man, and S. Kwong. Using multiobjective genetic algorithms to design mesh networks. *IEEE Computer*, 30(8):56 – 61, 1997.

[17] R. Kumar and N. Banerjee. Multicriteria network design using evolutionary algorithm. In *Proc. Genetic and Evolutionary Computations Conference (GECCO-03), LNCS 2023*, pages 2179 – 2190. Springer Verlag, 2003.

[18] R. Kumar, P. P. Parida, and M. Gupta. Topological design of communication networks using multiobjective genetic optimization. In *Proc. Congress Evolutionary Computation (CEC-2002)*, pages 425 – 430. IEEE Press, 2002.

[19] R. Kumar and P. I. Rockett. Improved Sampling of the Pareto-front in Multiobjective Genetic Optimization by Steady-State Evolution: A Pareto Converging Genetic Algorithm. *Evolutionary Computation*, 10(3):283 – 314, 2002.

[20] R. Kumar, P. K. Singh, and P. P. Chakrabarti. Multiobjective EA approach for improved quality of solutions for spanning tree problem. In *Proc. Int. Conf. Evolutionary Multi-Criterion Optimization (EMO-05), LNCS 3410*, pages 811 – 825, 2005.

[21] V. Paxson and S. Floyd. Wide-area traffic: The failure of Poisson modelling. *IEEE/ACM Trans. Networking*, 3(3):226 – 244, 1995.

[22] G. R. Raidl and B. A. Julstrom. Greedy heuristics and an evolutionary algorithm for the bounded-diameter minimum spanning tree problem. In *Proc. 18th ACM Symposium on Applied Computing (SAC 2003)*, pages 747 – 752, March 9-12 2003.

[23] 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.

[24] H.-P. Schwefel. Performance analysis of intermediate systems serving aggregated ON/OFF traffic with long-range dependent properties. PhD Thesis, Institute für Informatik: Technischen Universität München, 2000.

[25] G. Zohu and M. Gen. Genetic algorithm approach on multi-criteria minimum spanning tree problem. *European J. Operations Research*, 114(1):141–152, April 1999.