A Multicast Routing Algorithm Using Multiobjective Optimization

Jorge Crichigno, Benjamín Barán

P. O. Box 1439 - Universidad Nacional de Asunción
Asunción –Paraguay. Tel/Fax: (+595-21) 585619
{jcrichigno, bbaran}@cnc.una.py
http://www.una.py

Abstract. Multicast routing problem in computer networks, with more than one objective to consider, like cost and delay, is usually treated as a mono-objective Optimization Problem, where the cost of the tree is minimized subject to a priori restrictions on the delays from the source to each destination. This paper presents a new multicast algorithm based on the Strength Pareto Evolutionary Algorithm (*SPEA*), which simultaneously optimizes the cost of the tree, the maximum end-to-end delay and the average delay from the source node to each destination node. Simulation results show that the proposed algorithm is able to find Pareto optimal solutions. In addition, they show that for the problem of minimum cost with constrained end-to-end delay, the proposed algorithm provides better solutions than other well-known alternatives as *Shortest Path* and *KPP* algorithms.

1 Introduction

Multicast consists of concurrent data transmission from a source to a subset of all possible destinations in a computer network [1]. In recent years, multicast routing algorithms have become more important due the increased use of new point to multipoint applications, such as radio and TV transmission, on-demand video, teleconferences and e-learning. Such applications have an important quality-of-services (QoS) parameter, which is the end-to-end delay along the individual paths from the source to each destination. Another important consideration in multicast routing is the cost tree. It is given by the sum of the costs of its links. Most algorithms dealing with cost of a tree and delay from source to each destination, address multicast routing as a mono-objective optimization problem, minimizing the cost subjected to an end-to-end delay restriction. In [2], Kompella et al. present an algorithm (*KPP*) based on dynamic programming that minimizes the cost of the tree with a bounded end-to-end delay. For the same problem, Ravikumar et al. [3] present a method based on a simple genetic algorithm. This work was improved in turn by Zhengying et al. [4] and Araujo et al. [5]. The main disadvantage with this approach is the necessity of an a priory upper bound for the delay that may discard solutions of very low cost with a delay only slightly larger than a predefined upper bound. In contrast to the mono-objective algorithms, a *MultiObjective Evolutionary Algorithm*

(*MOEA*) simultaneously optimizes several objective functions; therefore, they can consider end-to-end delay as a new objective function. Multiobjective Evolutionary Algorithms provide a way to solve a multiobjective problem (*MOP*), finding a whole set of Pareto solutions in only one run [6]. This paper presents a new approach to solve the multicast routing problem based on a *MOEA* called the *Strength Pareto Evolutionary Algorithm* (*SPEA*) [6].

The remainder of this paper is organized as follow. A general definition of a multiobjective optimization problem is presented in Section 2. The problem formulation and the objective functions are given in Section 3. The proposed algorithm is explained in Section 4. Experimental results are shown in Section 5. Finally, the conclusions are presented in Section 6.

2 Multiobjective Optimization Problem

A general Multiobjective Optimization Problem (*MOP*) includes a set of n decision variables, k objective functions, and m restrictions. Objective functions and restrictions are functions of decision variables. This can be expressed as:

Optimize $\mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_k(\mathbf{x}))$
Subject to $\mathbf{e}(\mathbf{x}) = (e_1(\mathbf{x}), e_2(\mathbf{x}), ... ,e_m(\mathbf{x})) \geq \mathbf{0}$

where $\mathbf{x} = (x_1, x_2, ..., x_n) \in \mathbf{X}$ is the decision vector, and $\mathbf{y} = (y_1, y_2, ... , y_k) \in \mathbf{Y}$ is the objective vector. $\mathbf{X}$ denotes the decision space while the objective space is denoted by $\mathbf{Y}$. The set of restrictions $\mathbf{e}(\mathbf{x}) \geq \mathbf{0}$ determines the set of feasible solutions $\mathbf{X_f}$ and its corresponding set of objective vectors $\mathbf{Y_f}$. The problem consists of finding $\mathbf{x}$ that optimizes $\mathbf{f}(\mathbf{x})$. In general, there is no unique "best" solution but a set of solutions. Thus, a new concept of optimality is established for *MOP*s. Given $\mathbf{u}, \mathbf{v} \in \mathbf{X}$,

$\mathbf{f}(\mathbf{u})=\mathbf{f}(\mathbf{v})$ iff $\forall i \in \{1,2,...,k\}$: $f_i(\mathbf{u})=f_i(\mathbf{v})$;
$\mathbf{f}(\mathbf{u}) \leq \mathbf{f}(\mathbf{v})$ iff $\forall i \in \{1,2,...,k\}$: $f_i(\mathbf{u}) \leq f_i(\mathbf{v})$;
$\mathbf{f}(\mathbf{u})<\mathbf{f}(\mathbf{v})$ iff $\mathbf{f}(\mathbf{u}) \leq \mathbf{f}(\mathbf{v}) \wedge \mathbf{f}(\mathbf{u}) \neq \mathbf{f}(\mathbf{v})$.

Then, they comply with one of three conditions: $\mathbf{u}$ dominates $\mathbf{v}$ iff $\mathbf{f}(\mathbf{u})<\mathbf{f}(\mathbf{v})$; $\mathbf{u}$ and $\mathbf{v}$ are non-comparable iff $\mathbf{f}(\mathbf{u})\Box\mathbf{f}(\mathbf{v}) \wedge \mathbf{f}(\mathbf{v})\Box\mathbf{f}(\mathbf{u})$; and $\mathbf{v}$ dominates $\mathbf{u}$ iff $\mathbf{f}(\mathbf{v})<\mathbf{f}(\mathbf{u})$. $\mathbf{u}\Box\mathbf{v}$ will denote that $\mathbf{u}$ dominates or is equal to $\mathbf{v}$. A decision vector $\mathbf{x} \in \mathbf{X_f}$ is non-dominated with respect to a set $\mathbf{V} \subseteq \mathbf{X_f}$ iff: $\mathbf{x}$ dominates $\mathbf{v}$ or they are non-comparables, $\forall\, \mathbf{v} \in \mathbf{V}$. The set $\mathbf{X_{true}}=\{\mathbf{x} \in \mathbf{X_f} \mid \mathbf{x}$ is non-dominated with respect to $\mathbf{X_f}\}$ is known as Optimal Pareto set, while the corresponding set of objective vectors constitutes the Optimal Pareto Front.

3 Problem Formulation

For this work, a network is modeled as a direct graph $G=(V,E)$, where V is the set of nodes and E is the set of links. Let $(i,j) \in E$ be the link from node i to node j. For each link (i,j), let c_{ij} and d_{ij} its cost and delay. Let $s \in V$ denote a source and $N \subseteq V - \{s\}$ denote a set of destination nodes of a multicast group. Let $T(s,N)$ represent a multicast tree with s as source node and N as destination set. At the same time, let

$p_T(s,n)$ a subset of $T(s,N)$ that connects the source node s with a $n \in N$. The multicast routing problem may be stated as a *MOP* that tries to find a tree minimizing: 1-maximum delay (D_M); 2-cost of the tree (C); 3- average delay (D_A):

$$D_M = \underset{n \in N}{Max} \left\{ \sum_{(i,j) \in p_T(s,n)} d_{ij} \right\} . \quad \textbf{(1)}$$

$$C = \sum_{(i,j) \in T} c_{ij} . \quad \textbf{(2)}$$

$$D_A = \frac{1}{|N|} \sum_{n \in N} \left[\sum_{(i,j) \in p_T(s,n)} d_{ij} \right] . \quad \textbf{(3)}$$

Example 1. Given multicast group shown in Figure 1, a tree with an end-to-end delay less than 40 ms is a priori chosen. (a) shows the Shortest Path Tree (*SPT*). (b) shows the tree constructed with *KPP* [3], that minimizes *C* subject to the bound delay of 40ms. (c) shows a tree that would not be found by *KPP* or other algorithms based on restrictions if an a priori restriction of 40 ms were given. This alternative may be a good option since it has lower cost and a bound delay only slightly larger than the predefined bound. Note also that alternatives are non-dominated solutions.

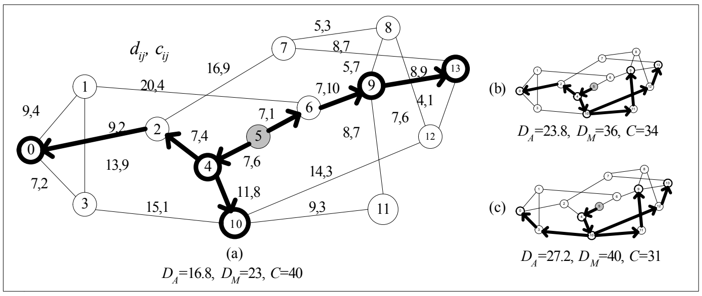


Fig. 1. National Science Foundation (NSF) Net. Each link with its delay and cost assigned

It is important to note from the mathematical formulation that the three objective functions are treated independently and should be minimized simultaneously. Therefore, a whole optimal solution set is provided in one run.

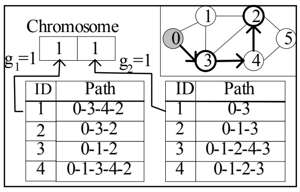
4 Proposed algorithm

The proposed algorithm holds an evolutionary population P and a Pareto set P_{nd}. Starting with a random population P, individuals evolve to optimal solutions, and these are included in P_{nd}. The algorithm, shown in Figure 2(a), is explained briefly.

Construct routing tables. Let $N = \{n_1, n_2, \ldots, n_{|N|}\}$. For each $n_i \in N$, a routing table is built. It consists of the R shortest and R cheapest paths. R is a parameter of the algorithm. Yen's algorithm [9] was used for this task. A chromosome is represented by a string of length $|N|$ in which the element (gene) g_i represents a path between s and n_i. The relation between a chromosome, genes and routing tables is shown in Figure 2(b). The chromosome represents the tree in the same Figure.

```
-Read group
-Construct routing tables
do{
   -Discard
   -Evaluate  P
   -Update non-dominated set  Pnd
   -Compute fitness
   -Selection
   -Crossover and mutation
} while stop criterion is not verified
```

(a)



(b)

Fig. 2. (a) Proposed algorithm. (b) Relation between the chromosome, genes and routing tables

Discard. In P, there may be duplicated chromosomes. Applying genetic operations like crossover on two of them will yield the same chromosome. Therefore, the searching ability could be reduced. Duplicated chromosomes are replaced by news randomly generated [8].

Evaluate P. Evaluate P computes the objective vector of each individual in P, using the objective functions defined in Section 3.

Update non-dominated set P_{nd}. Each non-dominated individuals of P is compared with the individuals in P_{nd}. If that in P is not dominated by anyone of P_{nd}, then it is copied to P_{nd}. Besides, if an individual in P_{nd} is dominated by someone in P, it is removed from the external set.

Compute fitness. Fitness is computed using the *SPEA* procedure [6].

Selection. The selection operator is applied on each generation over the union set of P_{nd} and P, to select good individuals to generate the next population P. The roulette procedure has been implemented as a selection operator [7].

Crossover and mutation. The two-point crossover operator is applied over each selected pair of individuals. Then, some genes in each chromosome of the new population are changed (mutated) with probability P_{mut} [7].

5 Results

The algorithm has been tested in different network topologies. Firstly, simulation experiments for the Example 1, labeled as P1, were performed. Besides, two test problems from [5] were used. They were labeled as P2 and P3. The parameters of the proposed algorithm were set to $|P|=100$, $P_{mut}=0.3$ and $R = 25$. An exhaustive search method, which finds the Pareto optimal solutions, was used to compare the results. This algorithm simply calculates all possible chromosomes and picks up the non-dominated individuals. The run time was approximately 3 hours for P1 and 10 minutes for P2 and P3. For each of the problems, 50 runs were done using the proposed algorithm with $|P|=50$, $P_{mut}=0.3$ and $R=25$. The runs were stop when no new non-dominated solution was found for 250 successive generations.

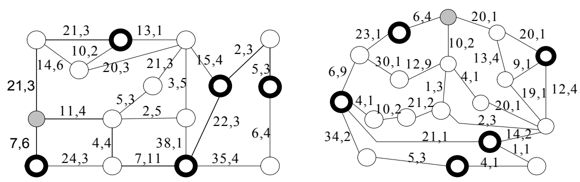


Fig. 3. Test problems P2 and P3

Table 1 presents the average (t_a) and maximum (t_{max}) running time in ms; the number of theoretical optimal solutions (*S)*; the minimum (SF_{min}), maximum (SF_{max}) and average (SF_a) number of theoretical optimal solutions found in a run. Note that even in the worst case the 66% of the theoretical Pareto set was found (4/6 for P3). Furthermore, the lower ratio SF_a/S was at least 0.857 (6/7, for P2).

Table 1. Results of problems P1, P2 and P3

	t_a	t_{max}	S	SF_{min}	SF_{max}	SF_a
P1	210	215	8	7	8	7.8
P2	350	390	7	5	7	6
P3	350	380	6	4	6	5.2

Besides the above test problems, the proposed algorithm was compared against the *SPT* and *KPP* [2] to find the minimum cost tree subject to an end-to-end delay restriction. The proposed algorithm was tested using the NSF Net. In order to measure the performance of the algorithm, average normalized cost and delay were computed:

$$C_N = (1/Y)\sum_{i=1}^{Y}\left(C_H^i - C_{MMA}^i\right)/\,C_{MMA}^i. \quad \textbf{(4)}$$

$$D_N = (1/Y)\sum_{i=1}^{Y}\left(D_H^i - D_{SPT}^i\right)/\,D_{SPT}^i. \quad \textbf{(5)}$$

where

Y : Number of runs with the same bound delay and size of multicast group.

C_H^i : Cost of the tree using H (H=*SPT* or H=*KPP*) on run *i*.

C_{MMA}^i : Cost of the tree using the proposed Multicast Multiobjective Algorithm (*MMA*) on run *i*.

D_H^i : Average delay of the tree using H (H=*MMA* or H=*KPP*) on run *i*.

D_{MMA}^i : Average delay of the *SPT*.

One hundred runs for each of four different multicast group sizes and for each of four different bound delays were done. Thus, 1600 runs with different multicast groups were tested. Costs of the links were generated at random and uniformly from the set {3, 4, …, 10} for each run. *MMA* was set to $|P|$=100, P_{mut}=0.3 and R= 30. The runs stopped when no new non-dominated solutions were found for 100 generations. Given that *MMA* provides more than one solution, the one showing the minimum cost subject to the end-to-end delay restriction was picked out. Figure 4 summarize the results. They show that *MMA* constructs lower cost trees than *KPP* and *SPT*. The normalized *SPT* costs are between 10 and 30 %, while those of *KPP* are between 2 and 5 %. The effect of increasing the bound delay is clear: *KPP* and *MMA* increase their average delay compared against *SPT*, while the cost of their trees are lower. This implies a notorious tradeoff between both metrics. Clearly an approach that can find Pareto solutions is much more suitable for this type of problems.

6 Conclusions

This paper presents a new multiobjective approach to solve the multicast routing problem. To solve this problem, a multiobjective multicast routing algorithm was proposed. This algorithm optimizes simultaneously three objective functions: 1-

maximum end-to-end delay, 2- cost of a tree, 3- average delay. The proposed evolutionary algorithm has a purely multiobjective approach, based on *SPEA*. This approach calculates not only one solution, but also an optimal Pareto set of solutions, in only one run. This last feature is of special importance, since the most adequate solution for each particular case can be chosen without a priori restrictions.

The proposed algorithm was evaluated with three test problems. Even in the worst case, it was able to find the 66% of the real Pareto set. Next, the proposed algorithm was compared against *SPT* and *KPP* to solve the problem of minimum cost tree subject to end-to-end delay restriction. Besides constructing the lowest cost tree, the proposed algorithm produces solutions with lower average delay than *KPP* in several cases (this is, cheaper trees with lower average delay), proving it is able to find better solutions including several theoretical Pareto optimal ones.

As future work, we will consider other objective functions as maximum link utilization and larger networks.

References

1. A. Tanenbaum, Computer Networks, Prentice Hall, 2003.
2. V. Kompella, J. Pasquale, and G. Polyzos, "Multicast routing in multimedia communication," IEEE/ACM Transactions on Networking, Vol. 1 No. 3, 1993, pp. 286-291.
3. C. P. Ravikumar, and R. Bajpai, "Source-based delay bounded multicasting in multimedia networks," Computer Communications, Vol. 21, 1998, pp. 126-132.
4. W. Zhengying, S. Bingxin, and Z. Erdun, "Bandwidth-delay-constraint least-cost multicast routing based on heuristic genetic algorithm," Computer Communications, Vol. 24, 2001, pp. 685-692.
5. P. T. de Araujo, and G. M. Barbosa, "Multicast Routing with Quality of Service and Traffic Engineering Requirements in the Internet, Based On Genetic Algorithm," Proceedings of the VII Brazilian Symposium on Neural Networks (SBRN´02), 2002.
6. E. Zitzler, and L. Thiele, "Multiobjective Evolutionary Algorithms: A comparative Case Study and the Strength Pareto Approach," IEEE Trans. Evolutionary Computation, Vol. 3, No. 4, 1999, pp. 257-271.
7. D. Goldberg, Genetic Algorithm is Search, Optimization & Machine Learning, Addison Wesley, 1989.
8. R.H. Hwang, W.Y. Do, and S.C. Yang, "Multicast Routing Based on Genetic Algorithms," Journal of Information Science and Engineering, Vol. 16, 2000, pp. 885-901.
9. J. Yen, "Finding the k shortest loopless path in a network," Management Science, 17:712-716, 1971.

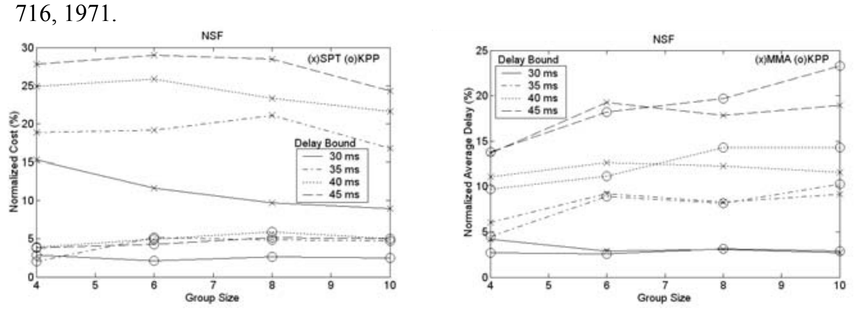


Fig. 4. Normalized cost and average delay