Multiobjective Multicast Routing Algorithm

Jorge Crichigno, Benjamín Barán

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

Abstract. This paper presents a new multiobjective multicast routing algorithm (*MMA*) based on the Strength Pareto Evolutionary Algorithm (*SPEA*), which simultaneously optimizes the cost of the tree, the maximum end-to-end delay, the average delay and the maximum link utilization. In this way, a set of optimal solutions, known as Pareto set, is calculated in only one run, without a priori restrictions. Simulation results show that *MMA* is able to find Pareto optimal solutions. They also show that for the constrained end-to-end delay problem in which the traffic demands arrive one by one, *MMA* outperforms the shortest path algorithm in maximum link utilization and total cost metrics.

1 Introduction

Multicast consists of concurrently 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, on-demand video and teleconferences. Such applications have an important quality-of-service (QoS) parameter, which is the end-to-end delay along the individual paths from the source to each destination.

Another consideration in multicast routing is the cost of the tree. It is given by the sum of the costs of its links. A particular case is given with unitary cost. In this case, a multicast tree with minimum number of links is preferred, such that bandwidth consumption is minimized. To improve the network resource utilization and to reduce hot spots, it is also important for a multicast routing algorithm to be able to balance traffic. In order to improve load balancing, minimization of the maximum link utilization is proposed [2].

Most algorithms deal with two of these metrics: cost of the tree and the end-to-end delay. They address the multicast routing as a mono-objective optimization problem, minimizing the cost subjected to a maximum end-to-end delay restriction. In [3], 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 to each destination. For the same problem, Ravikumar et al. [4] present a method based on a simple genetic algorithm. This work was improved in turn by Zhengying et al. [5] and Araujo et al. [6]. The main disadvantage with this approach is the necessity of an *a priory* upper bound for the end-to-end delay that may discard good solutions.

Lee et al. [2] present a multicast routing algorithm which finds a multicast tree minimizing the maximum link utilization subject to a hop-count constrained.

In contrast to the traditional mono-objective algorithms, a *MultiObjective Evolutionary Algorithm* (*MOEA*) simultaneously optimizes several objective functions; therefore, they can consider the maximum end-to-end delay, the average delay, the cost of the tree and the maximum link utilization as simultaneous objective functions. *MOEA*s provide a way to solve a multiobjective problem (*MOP*), finding a whole set of Pareto solutions in only one run [7]. This paper presents a Multiobjective Multicast Routing Algorithm (*MMA*), a new approach to solve the multicast routing problem based on a *MOEA* with an external population of Pareto Optimal solutions, called the *Strength Pareto Evolutionary Algorithm* (*SPEA*) [7].

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. **X** denotes the decision space while the objective space is denoted by **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 **x** that optimizes **f**(**x**). In general, there is no unique "best" solution but a set of solutions. Thus, a new concept of optimality should be established for *MOP*s. Given two decision vectors $\mathbf{u}, \mathbf{v} \in \mathbf{X}$,

$\mathbf{f}(\mathbf{u})=\mathbf{f}(\mathbf{v})$ 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: **u** dominates **v** iff $\mathbf{f}(\mathbf{u})<\mathbf{f}(\mathbf{v})$; **u** and **v** are non-comparable iff $\mathbf{f}(\mathbf{u}) \square \mathbf{f}(\mathbf{v}) \wedge \mathbf{f}(\mathbf{v}) \square \mathbf{f}(\mathbf{u})$; and **v** dominates **u** iff $\mathbf{f}(\mathbf{v}) < \mathbf{f}(\mathbf{u})$. $\mathbf{u} \square \mathbf{v}$ denotes that **u** dominates or is equal to **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: **x** dominates **v** or they are non-comparables, $\forall \mathbf{v} \in \mathbf{V}$. When **x** is non-dominated with respect to the whole set $\mathbf{X}_f$, it is called an optimal Pareto solution. The Pareto optimal set $\mathbf{X_{true}}$ may be defined as $\mathbf{X_{true}} = \{\mathbf{x} \in \mathbf{X_f} \mid \mathbf{x}$ is non-dominated with respect to $\mathbf{X_f}\}$. The corresponding set of objective vectors $\mathbf{Y}_{true}=\mathbf{f}(\mathbf{X}_{true})$ constitutes the Optimal Pareto Front.

3 Problem Formulation

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 $z(i,j)$, $c(i,j)$, $d(i,j)$ and $t(i,j)$ be its capacity, cost per bps, delay and current traffic, respectively. Let $s\in V$ denote a source, $N\subseteq V-\{s\}$ denote the set of destinations, and $\phi\in R^{+}$ the traffic demand (in bps) of a current multicast request. Let $T(s,N)$ represent a multicast tree with s as source node and N as destination set. At the same time, let $p_T(s,n)$ denote a path that connects the source node s with a destination node $n\in N$. Clearly, $p_T(s,n)$ is a subset of $T(s,N)$. The multicast routing problem may be stated as a *MOP* that tries to find a multicast tree that minimizes:

1- Maximum Delay:
$$D_M = \underset{n\in N}{Max}\left\{\sum_{(i,j)\in p_T(s,n)} d(i,j)\right\}. \tag{1}$$

2- Cost of the tree:
$$C = \sum_{(i,j)\in T} c(i,j). \tag{2}$$

3- Maximum link utilization:
$$\alpha_T = \underset{(i,j)\in T}{Max}\left\{\frac{\phi + t(i,j)}{z(i,j)}\right\}. \tag{3}$$

4- Average delay:
$$D_A = \frac{1}{|N|}\sum_{n\in N}\left[\sum_{(i,j)\in p_T(s,n)} d(i,j)\right]. \tag{4}$$

subject to
$$\phi + t(i,j) \le z(i,j) \quad \forall (i,j)\in T . \tag{5}$$

Example 1. Figure 1 shows the NSF network, with $d(i,j)$ in ms, $c(i,j)$, and $t(i,j)$ in Mbps. The capacity of the links is 1.5 Mbps. Suppose a traffic request arriving with ϕ=0.2 Mbps, s=5, and N={0, 4, 8, 9, 13}. (b) shows the tree constructed with *KPP* [3], subject to a maximum delay of 40 ms, (c) shows a tree would not be found by *KPP* or other algorithms based on restrictions if a bound delay lower than 40 ms were a priori established, even though it is a good option. If α_T is the most important metric, solution (d) would be the best alternative.

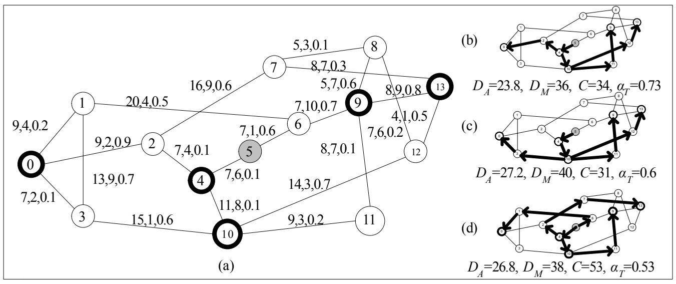


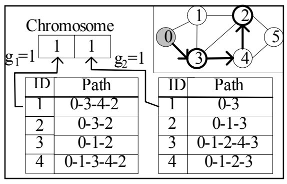
Fig. 1. The NSF Net. $d(i,j)$, $c(i,j)$ and $t(i,j)$ are shown over each (i,j) link. Different alternative trees for the multicast request with s=5, N={0, 4, 9, 10, 13} and ϕ=0.2 Mbps

4 Proposed algorithm

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

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

(a)



(b)

Fig. 2. (a) The proposed algorithm. (b) Relation between the chromosome, genes and routing tables. The chromosome represents the tree shown in the same Figure

Build routing tables: Let $N = \{n_1, n_2, ..., n_{|N|}\}$. For each $n_i \in N$, a routing table is built. It consists of the R shortest, R cheapest and R least used paths, where the use of a path is defined as the maximum link utilization along the path. R is a parameter of the algorithm. Yen's algorithm [10] 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. See Figure 2(b) to see the chromosome that represents the tree in the Figure.

Discard: In P, there may be duplicated chromosomes. Thus, duplicated chromosomes are replaced by new randomly generated individuals [9].

Evaluate: The individuals of P are evaluated using the objective functions. Then, non-dominated individuals of P are compared with the individuals in P_{nd} to **update the non-dominated set**, removing from P_{nd} dominated individuals.

Compute fitness: Fitness is computed using the *SPEA* procedure [7].

Selection: A roulette selection operator [8] is applied over the set $P_{nd} \cup P$ to generate the next population P.

Crossover and mutation: In this work, 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} [8].

5 Results

Simulation experiments were performed for Example 1. An exhaustive search method was used to compare the results. The optimal Pareto set was found to have 16 solutions. The run time of the exhaustive search method was approximately 3 hours.

One hundred runs were done using *MMA* with $|P|$ = 50, P_{mut} = 0.3, R = 25 and 500 generations. The minimum, maximum and average theoretical optimal solutions found by the runs using *MMA* were 16, 10 and 12.72 respectively. The mean running time was 270 ms and its maximum was 300 ms. Clearly *MMA* has a good performance finding at least 62.5% of the Pareto Front.

MMA was also compared against the delay shortest path (*SP*). Two hundred random requests of traffic demands of 0.1 Mbps were generated. The multicast group was randomly selected with a size between 4 and 7. The duration of each traffic demand was exponentially distributed (with an average of 120 s) and the inter-arrival time randomly distributed between 0 and 30 minutes. The maximum end-to-end delay for a group was set to 1.25 times the maximum end-to-end delay of the tree constructed with *SP*. *MMA* was set to $|P|$=100, P_{mut}=0.3 and R = 30. The mean time consumed to construct a multicast tree was 270 ms. Given that *MMA* may provide several solutions, two different scenarios were simulated: firstly, the trees with minimum α_T subject to end-to-end delay restriction; secondly, trees with minimum C satisfying the restriction. To compare performance, normalized values of maximum link utilization, total cost, which is calculated as the sum of the tree costs of the multicast groups already in the net, and the total delay, which is given by the sum of the total delay of the multicast groups already in the net, were calculated. For example, normalized total cost was given by $C_N = (C_{SP} - C_{MMA})/C_{MMA}$. Figure 3(a) shows that *MMA* leads to better link utilizations than *SP*. Note that the maximum link utilization using *SP* is sometimes 150 times greater than using *MMA*. Besides, from Figure 3(b), it can be seen that at almost all time, *SP* total cost is more expensive than *MMA*. As it was expected, the total delay using *SP* was lower than *MMA*, since *SP* produces optimal trees. Figures 4(a) to 4(c) show normalized values for the second scenario. Now the cost difference between *SP* and *MMA* has increased since the select criterion of *MMA* is the cost. Note that at almost all time, *SP* total cost is at least 10% more expensive than *MMA*. In contrast, α is even worse than with the first scenario.

6 Conclusion

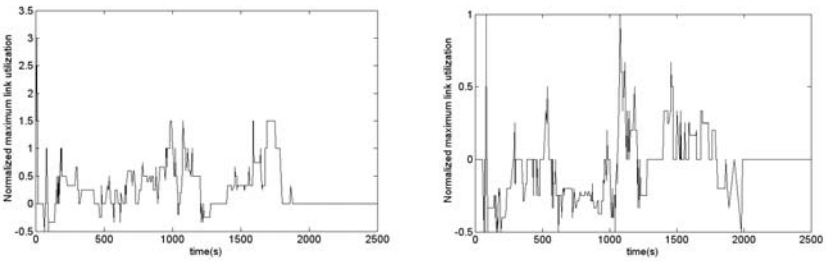
This paper presents a new multiobjective multicast routing algorithm (*MMA*) to solve the multicast routing problem. This new algorithm minimizes simultaneously four objective functions: 1- maximum end-to-end delay, 2- cost of a tree, 3- maximum link utilization and 4- average delay. *MMA* has a purely multiobjective approach, based on *SPEA*. This approach calculates an optimal Pareto set of solutions in only one run, without a priori restrictions, an importance feature of MMA.

Experimental results show that *MMA* was able to found Pareto optimal solutions. They also show that α and the total cost of *MMA* were lower than those of the shortest path algorithm. As future work, we will consider a traffic engineering scheme using different distribution trees over larger problems.

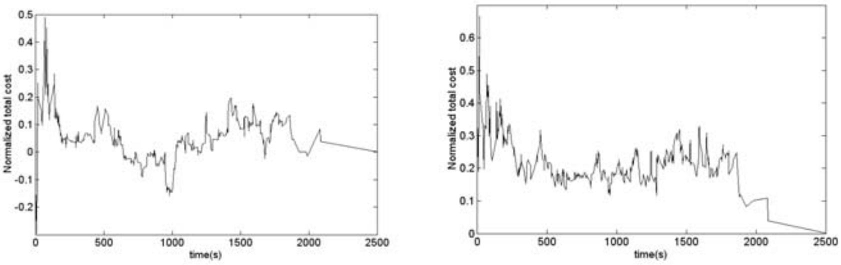
References

1. A. Tanenbaum, Computer Networks, Prentice Hall, 2003.
2. Y. Lee, Y. Seok, and Y. Choi, "Explicit Multicast Routing Algorithm for Constrained Traffic Engineering," Proc. of 7th Int. Symp. on Computer and Comm. (ISCC´02), 2002.

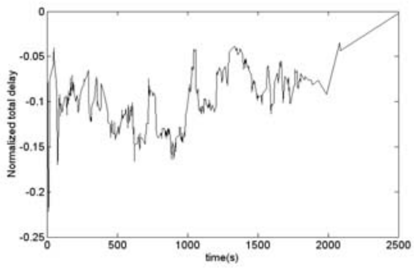
3. V. Kompella, J. Pasquale, and G. Polyzos, “Multicast routing in multimedia communication,” IEEE/ACM Transactions on Networking, Vol. 1 No. 3, 1993, pp. 286-291.
4. C. P. Ravikumar, and R. Bajpai, “Source-based delay bounded multicasting in multimedia networks,” Computer Communications, Vol. 21, 1998, pp. 126-132.
5. 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.
6. 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.
7. 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.
8. D. Goldberg, Genetic Algorithm is Search, Optimization & Machine Learning, Addison Wesley, 1989.
9. 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.
10. J. Yen, “Finding the k shortest loopless path in a network,” Management Science, 17:712-716, 1971.

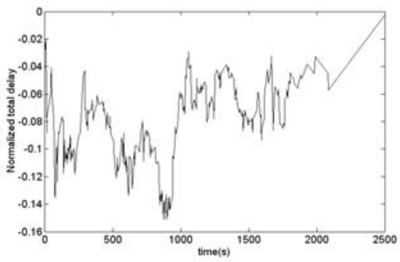


(a)



(b)





(c)

Fig. 3. Values for scenario 1

Fig. 4. Values for scenario 2