Multitree-Multiobjective Multicast Routing for Traffic Engineering

Joel Prieto[1], Benjamín Barán[1,2], Jorge Crichigno[1]

[1] Catholic University of Asunción. Tte. Cantaluppi y Villalón. PO Box 1638, Asunción, Paraguay
jprieto@telesurf.com.py, jcrichigno@ece.unm.edu
[2] National Computer Centre, National University of Asunción. PO Box 1439. San Lorenzo, Paraguay
bbaran@cnc.una.py

Abstract. This paper presents a new traffic engineering multitree-multiobjective multicast routing algorithm (M-MMA) that solves for the first time the GMM model for Dynamic Multicast Groups. Multitree traffic engineering uses several trees to transmit a multicast demand from a source to a set of destinations in order to balance traffic load, improving network resource utilization. Experimental results obtained by simulations using eight real network topologies show that this new approach gets trade off solutions while simultaneously considering five objective functions. As expected, when M-MMA is compared to an equivalent singletree alternative, it accommodates more traffic demand in a high traffic saturated network.

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 to the increased use of new point to multipoint applications, like radio and TV, on-demand video and e-learning. Such applications generally have some quality-of-service (QoS) requirements as maximum end-to-end delay and minimum bandwidth resources.

When a dynamic multicast problem considers various traffic requests, not only QoS parameters must be considered, but also load balancing and network resource utilization [2]. These objectives cannot be met by traditional *Best Effort* Internet routing approaches.

In order to solve this problem, Traffic Engineering proposes the optimization of network resources using load-balancing techniques. The main idea behind a load balancing technique for multicast transmission is to partition a data flow into several sub flows –or trees– between a source and all destination nodes. This objective is usually accomplished by minimizing the utilization (α) of the most heavily used network resource, as a link (what is known as *maximum link utilization*). Load balancing technique not only reduces hot spots over the network, but also provides the possibility of supporting connections of high bandwidth requirements through several links of low capacity.

Multicast Traffic Engineering problems (MTE) simultaneously consider several objectives to be optimized; therefore, it has been recognized as a Multiobjective Optimization Problem (MOP) [3]. A lot of multiobjective algorithms for multicast routing were proposed in the literature [3-6, 8-13, 15-18]. They are generalized in the *GMM model for Dynamic Multicast Groups* [11, 18]. GMM model considers a multitree multicast load-balancing problem with splitting in a multiobjective context.

This work presents a multitree routing algorithm that solves for the first time the dynamic problem of multicast routing considering not only static routing, but also dynamic routing, where multicast groups arrive one after another into a network.

The remainder of the document is organized as follows: Section 2 presents the mathematical formulation of the problem. A brief introduction to multiobjective optimization problems appears in Section 3. A complete explanation of the proposed algorithm is presented in Section 4. Testing scenarios are shown in Section 5. The experimental results are discussed in Section 6, while the final conclusions and future works are left for Section 7.

2 Problem Formulation

A network is modelled 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_{ij}, d_{ij} and $t_{ij} \in \Re^+$ be its capacity, delay and current traffic respectively. Let $s \in V$ denotes the source node, $N \subseteq V - \{s\}$ denote the set of destination nodes, and $\phi \in \Re^+$ the traffic demand (in kbps) of a multicast request, which is treated as a flow f. Let consider that f can be split into a number of sub flows f_k (k=1,2,..,$|K|$), where $|K|$ denotes the cardinality of set K. For each f_k, a multicast tree $T_k(s,N)$ must be constructed to transport a traffic ϕ_k , which is part of the total flow demand ϕ, as shown in (9).

Let $p_{Tk}(s, n) \subseteq T_k(s, N)$ denote the path that connects the source node s with a destination node $n \in N$ using tree T_k. Finally, let $d(p_{Tk}(s, n))$ and $h(p_{Tk}(s, n))$ represent the delay and the hop count of $p_{Tk}(s, n)$, i.e.,

$$d\left(p_{Tk}(s,n)\right) = \sum_{(i,j)\in p_{Tk}(s,n)} d_{ij} \qquad (1) \qquad\qquad h\left(p_{Tk}(s,n)\right) = \sum_{(i,j)\in p_{Tk}(s,n)} \mathbf{1} \qquad (2)$$

Using the above definitions, the multicast routing problem for traffic engineering treated in this paper is formulated as a MOP that tries to find a set of $|K|$ multicast trees $T_k(s,N)$ that minimizes the following five objective functions:

a- Maximal link utilization:

$$\alpha = \underset{\substack{(i,j)\in T_k \\ k\in K}}{Max} \left\{ (t_{ij} + \sum_{k=1}^{|K|} \phi_k\ )/z_{ij} \right\} \qquad (3)$$

b- Average delay:

$$D_A = \frac{1}{|N||K|} \sum_{k\in K} \sum_{n\in N} d\left(p_{Tk}(s,n)\right) \qquad (4)$$

c- Maximal delay:

$$D_M = \underset{\substack{n \in N \\ k \in K}}{Max} \left\{ d\left(p_{Tk}(s,n)\right) \right\} \quad \textbf{(5)}$$

d- Hop count average:

$$H_A = \frac{1}{|N||K|} \sum_{k \in K} \sum_{n \in N} h\left(p_{Tk}(s,n)\right) \quad \textbf{(6)}$$

e- Total bandwidth consumption:

$$BW = \sum_{k \in K} \phi_k \cdot |T_k| \quad \textbf{(7)}$$

subject to:

f- Link capacity constraint:

$$t_{ij} + \sum_{k \in K} \sum_{(i,j) \in Tk} \phi_k \le z_{ij} \quad \textbf{(8)}$$

g- Total information constraint:

$$\sum_{k=1}^{|K|} \phi_k = \phi \quad \textbf{(9)}$$

It should be mentioned that not all $|K|$ sub flows are necessary used. Therefore, if any $\phi_k = 0$ ($k = 1,2,..,|K|$), Eq. (4), (5) and (6) do not consider the corresponding $p_{Tk}(s, n)$ for calculation given that the tree is not used to transmit any information. Of course, the value of $|K|$ should be properly adjusted.

3 Multiobjective Optimization Problems

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

Optimize $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}) = (g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_r(\boldsymbol{x}))$.
Subject to $\boldsymbol{e}(\boldsymbol{x}) = (e_1(\boldsymbol{x}), e_2(\boldsymbol{x}), \ldots, e_c(\boldsymbol{x})) \ge \boldsymbol{0}$,
Where $\boldsymbol{x} = (x_1, x_2, \ldots, x_l) \in \boldsymbol{X}$ is the decision vector, and
$\boldsymbol{y} = (y_1, y_2, \ldots, y_r) \in \boldsymbol{Y}$ is the objective vector.

$\boldsymbol{X}$ denotes the decision space while the objective space is denoted by $\boldsymbol{Y}$. Depending on the problem at hand, "*optimize*" could mean minimize or maximize. The set of restrictions $\boldsymbol{e}(\boldsymbol{x}) \ge \boldsymbol{0}$ determines the set of feasible solutions $\boldsymbol{X_f}$ and its corresponding set of objective vectors $\boldsymbol{Y_f}$. A multiobjective problem consists in finding $\boldsymbol{x}$ that optimizes $\boldsymbol{g}(\boldsymbol{x})$. In general, there is no unique "best" solution but a set of solutions, none of which can be considered better than the others when all objectives are considered at the same time. This derives from the fact that there can be conflicting objectives. Thus, a new concept of optimality should be established for MOPs. Given two decision vectors $\boldsymbol{p}, \boldsymbol{q} \in \boldsymbol{X_f}$:

$\boldsymbol{g}(\boldsymbol{p}) = \boldsymbol{g}(\boldsymbol{q})$ iff $\forall i \in \{1, 2, \ldots, r\}$: $g_i(\boldsymbol{p}) = g_i(\boldsymbol{q})$
$\boldsymbol{g}(\boldsymbol{p}) \le \boldsymbol{g}(\boldsymbol{q})$ iff $\forall i \in \{1, 2, \ldots, r\}$: $g_i(\boldsymbol{p}) \le g_i(\boldsymbol{q})$
$\boldsymbol{g}(\boldsymbol{p}) < \boldsymbol{g}(\boldsymbol{q})$ iff $\boldsymbol{g}(\boldsymbol{p}) \le \boldsymbol{g}(\boldsymbol{q})$ and $\boldsymbol{g}(\boldsymbol{p}) \ne \boldsymbol{g}(\boldsymbol{q})$

Then, in a minimization context, two solutions $\boldsymbol{p}, \boldsymbol{q} \in \boldsymbol{X_f}$ satisfy one and only one of the following three conditions:

$\boldsymbol{p} \succ \boldsymbol{q}$ ($\boldsymbol{p}$ dominates $\boldsymbol{q}$), iff $\boldsymbol{g}(\boldsymbol{p}) < \boldsymbol{g}(\boldsymbol{q})$
$\boldsymbol{q} \succ \boldsymbol{p}$ ($\boldsymbol{q}$ dominates $\boldsymbol{p}$), iff $\boldsymbol{g}(\boldsymbol{q}) < \boldsymbol{g}(\boldsymbol{p})$
$\boldsymbol{p} \sim \boldsymbol{q}$ ($\boldsymbol{p}$ and $\boldsymbol{q}$ are non-comparable), iff $\boldsymbol{p} \nsucc \boldsymbol{q}$ and $\boldsymbol{q} \nsucc \boldsymbol{p}$.

A decision vector $\boldsymbol{x} \in \boldsymbol{X_f}$ is non-dominated with respect to a set $\boldsymbol{Q} \subseteq \boldsymbol{X_f}$ iff: $\boldsymbol{x} \succ \boldsymbol{q}$ or $\boldsymbol{x} \sim \boldsymbol{q}$, $\forall\, \boldsymbol{q} \in \boldsymbol{Q}$. When $\boldsymbol{x}$ is non-dominated with respect to the whole set $\boldsymbol{X_f}$, it is called an optimal Pareto solution; therefore, the *Pareto optimal set* $\boldsymbol{X_{true}}$ may be formally defined as: $\boldsymbol{X_{true}} = \{\boldsymbol{x} \in \boldsymbol{X_f} \mid \boldsymbol{x}$ is non-dominated with respect to $\boldsymbol{X_f}\}$. The corresponding set of objective vectors $\boldsymbol{Y_{true}} = \boldsymbol{f}(\boldsymbol{X_{true}})$ constitutes the *Optimal Pareto Front.*

4 Proposed Algorithm

Inspired in the SPEA scheme [14] the proposed M-MMA algorithm holds an evolutionary population P and an external Pareto solution set P_{nd}. The algorithm begins with a set of random configurations called initial population. Each individual in the population represents a potential solution to the problem.

At each generation, the individuals are evaluated using an adaptability function, also known as *fitness*, proposed by SPEA, which is based on the dominance criterion presented in section 3. Based on this value, some individuals called parents are selected. The probability of selection of an individual is related to its *fitness*. Then, genetic probabilistic operators are applied to the parent to construct new individuals that will be part of a new population. The process continues until a stop criterion (as a maximum number of generations) is satisfied. M-MMA is summarized in Fig. 1.

- Read multicast group and traffic demand
- Initialize sets P and P_{nd}
Do {
- Discard identical individuals of P
- Calculate objective functions of each individual in P
- Apply local optimization algorithm
- Update non-dominated set P_{nd}
- Compute fitness
- Selection
- Construct new solutions using *crossover*
} while stop criterion is not satisfied

Fig. 1. M-MMA algorithm

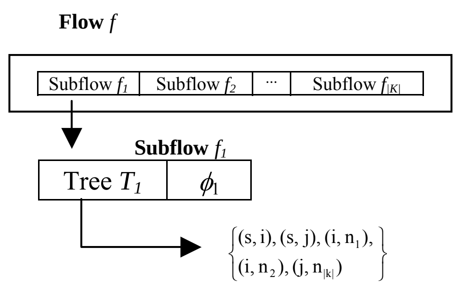

Fig. 2. Chromosome representation

4.1. Encoding

Each chromosome or individual is a candidate solution for the problem. Inspired in the GMM-model [11], an individual is represented by a set of trees transporting a flow f (Fig. 2). Each flow is split in $|K|$ sub flows, as shown in (9), with a tree T_k transmitting sub flow f_k. A tree is represented by the set of links belonging to it [6]. The field ϕ_k associated to each sub flow is the total information transmitted through T_k. This encoding scheme was selected motivated by the promising results obtained by Crichigno *et al.* [6], who conclude that better solutions are found when the trees are represented as a set of links instead of different paths.

4.2. Initial population

The procedure proposed in M-MMA to generate each initial solution of P is shown in Fig. 3. The initialization procedure, called PrimRST (Prim Random Steiner

Tree), was proposed in [6]. Starting with a source node *s*, at each iteration, the algorithm expands the tree T_k by choosing a new link from a set *A*, which contains all possible new links for the tree. A set V_c contains the nodes already in the tree. The procedure continues until all destination nodes *N* are included in V_c. The value of ϕ_k is initialized as $\phi / |K|$. The value of $|K|$ should be previously decided by the traffic engineer. For the experimental results that follows, $|K| = 2$ was chosen. We have considered this small value because the problem is very complex. Moreover, in GMM model [11] the quantity of sub flows is considered as an objective function, because this algorithm is thought for MPLS networks [2], where the quantity of labels is limited. The PrimRST algorithm is iteratively used to construct each tree T_k of the $|K|$ trees that constitute a chromosome, as shown in Fig. 2.

```
PrimRST(G(V,E), s, N)
- T_k = {};
- V_c = {s};
- A = {(s, j) | (s, j) ∈ E, j ∈ V};
do{
      - Choose a link (i, j) ∈ A at random.
      - A = A – {(i, j)}.
       If j ∉ V_c Then
           - T_k = T_k ∪ {(i, j)}.
           - V_c = V_c ∪ {j}.
           - A = A ∪ {(j, w) | (j, w) ∈ E , w ∉ V_c};
       End if
} while (N ∪ {s} ⊄ V_c)
- Prune useless links of T_k
- Return T_k
```

Fig. 3. Procedure PrimRST used to build random multicast Trees

```
Local Optimization (P, Δ0, ε )
For i=1 until |P|
  Δ=Δ0
  While Δ > ε
       If φ1^i + Δ·φ^i ≤ φ^i then
            φ1* = φ1^i + Δ·φ^i
            φ2* = φ^i - φ1^i
            Evaluate individual f*
            If f* ≻ f^i then
                 φ1^i = φ1*
                 φ2^i = φ2*
            else
                 Δ = Δ / 2
            End if
       else
            Δ = Δ / 2
       End if
  End while
End for
```

Fig. 4. Local Optimization Procedure

4.3. Local optimization

This procedure tries to optimize the amount of information ϕ_k to be transmitted through each sub flow, satisfying (8) and (9). In order to differentiate between two individuals of *P*, let f^i be the *i*-th flow or individual of *P* (i=0,1,…,|*P*|), ϕ^i the total flow demand for that individual, and ϕ_k^i the *k*-information amount transmitted through sub flow f_k^i. Local Optimization procedure is presented in Figure 4. The process modifies the values of ϕ_k^i in the following way:

a) ϕ_1^i is increased and ϕ_2^i is decreased in a percentage Δ of ϕ^i. In fact, ϕ_2^i is calculated as (ϕ^i - ϕ_1^i). Initial value for Δ (known as Δ_0) and its minimum value ε are parameters of the procedure.
b) If total information constraint (9) is fulfilled, new temporal values ϕ_1^* and ϕ_2^* are calculated and objective vector f^* is evaluated; otherwise, Δ is reduced to Δ/2 and the process goes back to step a).
c) If the new solution f^* dominates f^i, new values ϕ_1^* and ϕ_2^* are accepted as current best value and the process continues; otherwise, Δ is reduced to Δ/2 and

the procedure goes back to step a).

d) Iteration continues while $\Delta > \varepsilon$.

Once the iteration is completed, a new iteration begins, but instead of incrementing ϕ_1^i, it is decreased.

4.4. Crossover

The crossover algorithm is based on the one originally presented by Zhengying *et al.* [16]. It was also used in several other publications [6, 7, 15]. The algorithm has four stages:

1. choose one tree from each parent;
2. identify common links of the selected pair. These links will be part of the child tree that will be in the next generation of *P*. Given that common links of the parents could lead to a child composed of disjoined sub-trees, new links may be added [16];
3. connect the disjoined sub-trees until a multicast tree is constructed. At this step, the sub-trees are connected at random. Each sub-tree has a root node. At each iteration, an interconnection algorithm adds a new link, which has a source-node already in a sub-tree. Two sub-trees are connected when the root of one sub-tree (T_1) is the destination node of the selected link, and the source node of the link belongs to the other sub-tree (T_2); the root of the new sub-tree is the root of T_2;
4. calculate $\phi_j = \left(\phi_j^p + \phi_j^q\right)/2$, for both sub flows j=1, 2, where ϕ_j^p and ϕ_j^q are the j-information amount (ϕ_j) from the two parent trees *p* and *q*.

In order to fulfil the flow constraint given by (9), a normalized process computing ϕ_k is used. For a new individual, the new ϕ_k is given by the following equation:

$$\phi_k^{new} = \phi\left(\phi_k \Big/ \sum_{k=1}^{|K|} \phi_k\right) \quad \mathbf{(10)}$$

5 Testing Scenario

Eight network topologies were used for testing purpose. They were: NTT (Nippon Telephone and Telegraph Co., Japan) [5], NSF (National Science Foundation, United States-US) [5], Telstra (Australia) [19], Sprintlink (US) [19], Ebone (Europe) [19], Tiscali (Europe) [19], Exodus (US) [19] and Abovenet (US) [19].

In order to compare M-MMA behaviour under several traffic loads over the network, three scenarios were defined for every topology: (a) low load, (b) high load and (c) saturation. For every scenario, Ψ traffic requests were generated, simulating a dynamic situation in which they arrive one after another. Each traffic request was created using a *groupGenerator* algorithm [7], summarized in Fig. 5.

The *groupGenerator* algorithm generates a multicast group with a destination size between $|N|_{min}$ and $|N|_{max}$; then, *random*(unif, 0, 2000) gives the arrival time of the group, with a uniform distribution between 0 and 2000 seconds. The duration of each group was exponentially distributed, with an average of 60 seconds. Finally,

```
groupGenerator
group(i)  =   groupGenerator(|N|min, |N|max);
Tbeg(i)   =   random(unif, 0,2000);
Tend(i)   =   Tbeg(i) + random(exp, 0,2000);
φ(i)      =   random(unif, φmin, φmax);
End groupGenerator
```

Fig. 5. GroupGenerator algorithm

the traffic demand is set to a value between ϕ_{min} and ϕ_{max}. The parameters used to generate each scenario are given in Table 1.

Talavera *et al.* [7] showed that most MOEAs may suit for the task of routing multicast demand, but the main factor to define performance in a dynamical environment is the policy used to choose a specific solution from a Pareto front. They proposed different policies to perform this task, proving that the policy of choosing the closest solution to the origin provides excellent trade-off values, outperforming the traditional policy of choosing the solution with better α. Consequently, we use that approach to select a solution from a Pareto front in our experiments. It is useful to mention that [7] concluded that average number of rejected groups might be considered an important metric to compare different algorithms and policies.

Table 1. Parameters used to generate testing scenarios

Network topology			Scenarios load	Parameters				
Name (Location)	Nodes	Links		Ψ	$\lvert N\rvert_{min}$	$\lvert N\rvert_{max}$	ϕ_{min}	ϕ_{max}
Telstra (Australia)	57	118	Low	200	4	10	25	50
			High	300	10	25	50	200
			Saturation	400	10	35	75	300
Sprintlink (US)	44	166	Low	200	3	6	25	50
			High	300	9	12	50	200
			Saturation	400	9	20	75	300
Ebone (Europe)	23	76	Low	200	3	6	25	50
			High	300	5	10	50	200
			Saturation	400	8	15	75	300
Tiscali (Europe)	49	172	Low	200	4	6	25	50
			High	300	9	12	50	200
			Saturation	400	10	20	75	300
Exodus (US)	22	74	Low	200	3	6	25	50
			High	300	5	10	50	200
			Saturation	400	8	15	75	300
Abovenet (US)	33	84	Low	200	3	6	25	50
			High	300	5	10	50	200
			Saturation	400	8	15	75	300
NTT (Japan)	55	144	Low	200	4	10	100	200
			High	300	10	25	200	800
			Saturation	400	10	35	200	800
NSF (US)	14	42	Low	200	2	5	25	50
			High	300	3	7	50	200
			Saturation	400	6	9	75	300

For this problem, M-MMA was compared against MMA2 algorithm [6]. MMA2 is a multiobjective multicast algorithm that routes a request demand through only one tree. We have chosen this algorithm because of its promising results when compared to other alternatives as MMA1 [4, 5] and SK [17]. The following dominance metrics were taken into account:

D_{MMA2}: Percentage of solutions selected using MMA2 that dominates the corresponding M-MMA solutions.

$D_{M\text{-}MMA}$: Percentage of solutions selected using M-MMA that dominates the corresponding MMA2 solutions.

I: Percentage of indifference relationships. This occurs when solutions found by MMA2 and M-MMA are non-comparables.

Eq: Percentage of solutions found by both algorithms that have equal values for objective functions.

We also have compared the amount of solutions selected by M-MMA that uses only one tree to transmit the traffic demand. Finally, percentages of rejected groups for lack of link capacity are given for each scenario.

6 Experimental results

Results for the simulations performed on eight network topologies are shown in tables 2, 3 and 4.

Table 2 summarizes the amount of solutions for each scenario according to the dominance metrics defined in section 5. There is not a clear dominant algorithm, given that many solutions are indifferent (in a multiobjective context) or they have identical values for the objective vectors. Shadowed cells in table 2 highlight this fact. This result is not a surprise, given that we are considering several conflicting objective functions.

Table 2. Classification of solutions according to dominance metrics

Network	Scenario	D_{MMA2}	$D_{M\text{-}MMA}$	*I*	*Eq*
Telstra	Low	32.50	5.50	50.50	11.50
	High	11.33	17.67	57.33	13.67
	Saturation	26.00	8.75	48.00	17.25
Sprintlink	Low	3.00	11.50	2.50	83.00
	High	3.67	16.33	0.67	79.33
	Saturation	21.50	11.75	14.25	52.50
Ebone	Low	7.50	13.00	3.00	76.50
	High	21.00	12.67	3.33	63.00
	Saturation	7.50	10.75	29.25	52.50
Tiscali	Low	7.00	13.50	3.00	76.50
	High	2.33	0.00	3.67	94.00
	Saturation	9.25	0.75	28.50	61.50
Exodus	Low	8.50	2.50	4.00	85.00
	High	9.33	7.67	1.00	82.00
	Saturation	6.50	12.00	10.25	71.25
Abovenet	Low	11.50	4.50	5.50	78.50
	High	2.67	10.33	1.00	86.00
	Saturation	11.75	12.50	22.50	53.25
NTT	Low	0.50	34.50	0.50	64.50
	High	2.33	27.67	0.33	69.67
	Saturation	5.50	22.50	2.75	69.25
NSF	Low	4.50	9.50	6.50	79.50
	High	0.67	10.33	0.00	89.00
	Saturation	8.00	15.50	2.75	73.75

The percentage of multicast groups routed by a single tree is given in Table 3. We should clarify that M-MMA solutions not always use multitree, given that one tree may transport the whole information ϕ. In many cases, both algorithms found the same unitree solution. Multitree solution is used only when it is clearly better than unitree. This is the main reason why M-MMA could find better global solutions.

Actually, a mean of 63.2% of the best solutions had only one tree, and M-MMA is able to find those solutions, just as MMA2. However, in several opportunities the best solution for a given situation is multitree and therefore, only M-MMA is able to find it, making clear why M-MMA outperforms MMA2.

Finally, table 4 gives an idea about multitree performance considering the percentage of rejected groups for lack of link capacity. This result illustrates that M-MMA solutions fulfil the Traffic Engineering purpose, using load-balancing techniques in order to optimize network resources, and therefore, accommodating more traffic than a purely unitree approach like MMA2.

Table 3. Percentage of multicast groups routed by a single tree

Network	Scenario	%
Telstra	Low	93.50
	High	91.67
	Saturation	58.25
Sprintlink	Low	53.00
	High	85.00
	Saturation	83.75
Ebone	Low	46.00
	High	68.33
	Saturation	57.25
Tiscali	Low	82.50
	High	82.00
	Saturation	83.00
Exodus	Low	41.50
	High	53.00
	Saturation	60.25
Abovenet	Low	50.50
	High	77.33
	Saturation	74.50
NTT	Low	25.00
	High	28.33
	Saturation	46.50
NSF	Low	62.00
	High	50.00
	Saturation	42.00

Table 4. Percentage of groups rejected for lack of link capacity for both algorithms

Network	Scenario	% Rejected by	
		MMA2	M-MMA
Telstra	Low	0.00	0.00
	High	5.67	5.67
	Saturation	37.75	35.75
Sprintlink	Low	0.00	0.00
	High	0.33	0.00
	Saturation	14.00	9.00
Ebone	Low	0.00	0.00
	High	2.00	2.00
	Saturation	27.00	27.00
Tiscali	Low	0.00	0.00
	High	3.00	0.00
	Saturation	28.50	7.75
Exodus	Low	0.00	0.00
	High	0.00	0.00
	Saturation	10.25	10.00
Abovenet	Low	0.00	0.00
	High	0.00	0.00
	Saturation	22.00	19.50
NTT	Low	0.00	0.00
	High	0.33	0.00
	Saturation	2.50	1.50
NSF	Low	0.00	0.00
	High	0.00	0.00
	Saturation	1.75	1.25

7 Conclusion and future work

This paper presents the M-MMA algorithm, which is able to solve for the first time the GMM-model in a dynamical environment, considering multitree. The proposed algorithm treats the multiobjective problem of multicast routing in a network, splitting traffic demand into several trees (multitree context) to optimize network resource utilization. To better accomplish the optimization goal, M-MMA proposes a local optimization procedure that finds better solutions improving the relative amount of information to be transmitted through each tree.

Results obtained by simulations on dynamical environments where traffic demands come one after another show that no studied algorithm is clearly dominant. In fact, many times the best solution under the given policy had only one tree; however, the best solution for a given situation is sometimes a multitree and therefore, only

M-MMA is able to find it. As a consequence, M-MMA is able to accommodate more traffic demand under a saturated scenario. For further study, we plan to consider simultaneous routing of several multicast requests in optical networks.

References

1. A. Tanenbaum: *Computer Networks*, Prentice Hall, 2003.
2. D. Awdeuche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus: *Requirements For Traffic Engineering Over MPLS*. RFC 2702. 1999.
3. J. Crichigno, and B. Barán: *Multiobjective Multicast Routing Algorithm*. IEEE ICT'2004, Ceará, Brazil, 2004.
4. B. Barán, and J. Crichigno: *A Multicast Routing Algorithm Using Multiobjective Optimization*. IEEE ICT'2004, Ceará, Brazil, 2004.
5. J. Crichigno, and B Barán: *Multiobjective multicast routing algorithm for traffic engineering*. IEEE ICCCN 2004, Chicago USA.
6. J. Crichigno, F. Talavera, J. Prieto, and B. Barán: *Enrutamiento Multicast utilizando Optimización Multiobjetivo*. CACIC'2004, Buenos Aires, Argentina, 2004. pp. 147-158.
7. F. Talavera, J. Crichigno, and B. Barán: *Policies for Dynamical MultiObjective Environment of Multicast Traffic Engineering*. IEEE ICT 2005, South Africa.
8. Y. Donoso, R. Fabregat, and J. Marzo: *Multi-Objective Optimization Algorithm for Multicast Routing with Traffic Engineering*. IEEE ICN 2004.
9. R. Fabregat, Y. Donoso, J.L. Marzo, and A. Ariza: *A Multi-Objective Multipath Routing Algorithm for Multicast Flows*. SPECTS 2004.
10. R. Fabregat, Y. Donoso, F. Solano, and J.L. Marzo: *Multitree Routing for Multicast Flows: A Genetic Algorithm Approach*. CCIA 2004.
11. Y. Donoso, R. Fabregat, F. Solano, J. L. Marzo, and B. Barán: *Generalized Multiobjective Multitree model for Dynamic Multicast Groups*. IEEE ICC 2005, Seul Corea.
12. A. Roy, N. Banerjee, and S. Das: *An efficient Multi-Objective QoS-Routing Algorithm for Wireless Multicasting*. INFOCOM 2002.
13. X. Cui, C. Lin, and Y. Wei: *A Multiobjective Model for QoS Multicast Routing Based on Genetic Algorithm*. ICCNMC 2003.
14. 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.
15. P. Texeira de Araújo, and G. Barbosa Oliveira: *Algoritmos Genéticos Aplicados al Ruteamiento Multicast en Internet, Contemplando Requisitos de Calidad de Servicio e Ingeniería de Tráfico*. VII Brazilian Symposium on Neural Networks (SBRN´02), 2002.
16. W. Zhengying, S. Bingxin, and Z. Erdun: *Bandwidth-delay-constraint least cost multicast routing based on heuristic genetic algorithm*. Computer Communications. 2001. Vol. 24. pp. 685-692.
17. Y. Seok, Y. Lee, Y Choi, and C. Kim: *Explicit Multicast Routing Algorithm for Constrained Traffic Engineering*. IEEE ISCC´02. Italia, 2002.
18. R. Fabregat, Y. Donoso, B. Barán, F. Solano, and J.L. Marzo: *Multi-objective Optimization Scheme for Multicast Flows: a Survey, a Model and a MOEA Solution*. IFIP/ LANC 2005.
19. Spring, R Mahajan, and D. Wetheral: *Measuring ISP topologies with Rocketfuel*. Proceedings of the ACM SIGCOMM´02 Conference, 2002.