Solving Multiobjective Multicast Routing Problem with a new Ant Colony Optimization approach

Diego Pinto
National University of Asunción
Catholic University of Asunción
(585 21) 585-559, P.O. Box. 1439 - Paraguay
dpinto@cnc.una.py

Benjamín Barán
National University of Asunción
Catholic University of Asunción
(585 21) 585-559, P.O. Box. 1439 - Paraguay
bbaran@cnc.una.py

ABSTRACT
This work presents two multiobjective algorithms for Multicast Traffic Engineering. The proposed algorithms are new versions of the Multi-Objective Ant Colony System (MOACS) and the Max-Min Ant System (MMAS), based on Ant Colony Optimization (ACO). Both ACO algorithms simultaneously optimize maximum link utilization and cost of a multicast routing tree, as well as average delay and maximum end-to-end delay, for the first time using an ACO approach. In this way, a set of optimal solutions, know as Pareto set is calculated in only one run of the algorithms, without a priori restrictions. Experimental results show a promising performance of both proposed algorithms for a multicast traffic engineering optimization, when compared to a recently published Multiobjective Multicast Algorithm (MMA), specially designed for Multiobjective Multicast Routing Problems.

Categories and Subject Descriptors
C.2.2 [**Computer-Communication Networks**]: Network Protocols – *Routing protocols.*

General Terms
Algorithms and Experimentation.

Keywords
Traffic Engineering, Ant Colony Optimization, Multicast Routing, Multiobjective Optimization.

1. INTRODUCTION
Multicast consists of simultaneous data transmission from a source node to a subset of destination nodes in a computer network [1]. Multicast routing algorithms have recently received great attention due to increased use of recent point-to-multipoint applications, such as radio and TV transmission, on-demand video, teleconferences and so on. Such applications generally require several quality-of-service (QoS) parameters such as maximum end-to-end delay and minimum bandwidth resources, subject to optimization with a traffic-engineering scheme.

LANC'05, October 10-12, 2005, Cali, Colombia.

When a dynamic multicast problem considers several traffic requests, not only QoS parameters must be considered, but also load balancing and network resources utilization must be taken into account. In order to avoid hot spots and to balance the network load, a common approach is to minimize the utilization of the most heavily used link in the network or maximum link utilization [2]. Therefore, cost minimization of the tree of each multicast group, which is given by the sum of the cost of the used links, is also desired. It is known that the complexity of computing the minimum cost tree for a given multicast group is NP-hard [3]. Then, this paper presents new ACO versions of the Multi-Objective Ant optimization System (MOACS) [4] and the Max-Min Ant System (MMAS) [5], finding a set of optimal solutions by simultaneously optimizing four objective functions: (1) maximum link utilization, (2) cost of the multicast tree, (3) maximum end-to-end delay and (4) average delay. In this way, a whole Pareto set of optimal solutions can be obtained in only one run on the proposed algorithms. For theoretical studies a whole Pareto set is computed. The selection of one solution of this Pareto set is studied in [6].

To verify the performance of the proposed algorithms, simulations were carried out with different sizes of multicast groups on diverse topology networks. The two proposed algorithms were compared to a Multiobjective Optimization Evolutionary Algorithm (MOEA) specially designed to solve that multicast routing problem, the recently published Multicast Multiobjective Algorithm (MMA) [7, 8, 9] based on the Strength Pareto Evolutionary Algorithm (SPEA) [10].

The remainder of this work is organized as follows. Section 2 describes related publications. A general definition of a multiobjective problem is presented in Section 3. The problem formulation and the objective functions are given in Section 4. Ant Colony Optimization approach is explained in Section 5. The two proposed algorithms are explained in Section 6 and Section 7, respectively. The Multicast Multiobjective Algorithm (MMA) is summarized in Section 8 while the experimental environment is shown in Section 9. Section 10 presents experimental results. Finally, conclusions and future works are left for Section 11.

2. RELATED WORK
Several algorithms based on ACO consider the multicast routing as a mono-objective problem, minimizing the cost of the tree under multiple constraints. In [11] Liu and Wu propose the

construction of a multicast tree, where only the cost of the tree is minimized using a degree constraints. On the other hand, Gu et al. considered multiple parameters of QoS as constraints, minimizing just the cost of the tree [12]. It can be clearly noticed that previous algorithms treat the Multicast Traffic Engineering problem as a mono-objective problem with several constraints. The main disadvantage of these approaches is the necessity of an *a priori* predefined upper bound that can exclude good practical solutions.

In [13], Donoso et al. proposed a multi-tree traffic-engineering scheme using multiple trees for each multicast group. They took into account four metrics: maximum link utilization (α), hop count, bandwidth consumption and total end-to-end delay. The method minimizes a weighted sum function composed of the above four metrics. Considering the scheme is NP-hard, the authors proposed a heuristic algorithm consisting of two steps:

1**.** Obtaining a modified graph, where all possible paths between the source node and every destination node are looked for.
2**.** Finding out the solution trees, based on the distance values and the available capacity of the paths, in the modified graph.

Recently, Crichigno and Barán [7, 8, 9] have proposed a Multiobjective Multicast Algorithm (MMA), based on the Strength Pareto Evolutionary Algorithm (SPEA) [10], which simultaneously optimizes maximum link utilization, cost of the tree, maximum end-to-end delay and average delay. The MMA algorithm finds a set of optimal solutions, which is calculated in only one run, without *a priori* restrictions; therefore, it will be used in this paper as a reference for comparison.

3. MULTIOBJECTIVE OPTIMIZATION PROBLEMS

A general Multiobjective Optimization Problem (MOP) [14] 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	$\boldsymbol{y = f(x) = (f_1(x), f_2(x), \dots, f_k(x))}$.
Subject to	$\boldsymbol{e(x) = (e_1(x), e_2(x), \dots, e_m(x)) \geq 0}$,
where	$\boldsymbol{x = (x_1, x_2, \dots, x_m)} \in \boldsymbol{X}$ *is the decision vector,*
and	$\boldsymbol{y = (y_1, y_2, \dots, y_k)} \in \boldsymbol{Y}$ *is the objective vector.*

$\boldsymbol{X}$ denotes the decision space while the objective space is denoted by $\boldsymbol{Y}$. Depending on the kind of the problem, "*optimize*" could mean minimize or maximize. The set of restrictions $e(x) \geq 0$ determines the set of feasible solutions $\boldsymbol{X_f} \subseteq \boldsymbol{X}$ and its corresponding set of objective vectors $\boldsymbol{Y_f} \subseteq \boldsymbol{Y}$. A multiobjective problem consists in finding $\boldsymbol{x}$ that optimizes $\boldsymbol{f}(\boldsymbol{x})$. 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 $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{X_f}$:

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

Then, in a minimization context, $\boldsymbol{u}$ and $\boldsymbol{v}$ comply with one and only one of the following three conditions:

$\boldsymbol{u} \succ \boldsymbol{v}$ ($\boldsymbol{u}$ dominates $\boldsymbol{v}$), iff: $\boldsymbol{f}(\boldsymbol{u}) < \boldsymbol{f}(\boldsymbol{v})$
$\boldsymbol{v} \succ \boldsymbol{u}$ ($\boldsymbol{v}$ dominates $\boldsymbol{u}$), iff: $\boldsymbol{f}(\boldsymbol{v}) < \boldsymbol{f}(\boldsymbol{u})$
$\boldsymbol{u} \sim \boldsymbol{v}$ ($\boldsymbol{u}$ and $\boldsymbol{v}$ are non-comparable),
iff: $\boldsymbol{f}(\boldsymbol{u}) \nless \boldsymbol{f}(\boldsymbol{v}) \wedge \boldsymbol{f}(\boldsymbol{v}) \nless \boldsymbol{f}(\boldsymbol{u})$

Alternatively, for the rest of this work, $\boldsymbol{u} \triangleright \boldsymbol{v}$ will denote that $\boldsymbol{u}$ dominates or is equal to $\boldsymbol{v}$. A decision vector $\boldsymbol{x} \in \boldsymbol{X_f}$ is non-dominated with respect to a set $\boldsymbol{Q} \subseteq \boldsymbol{X_f}$ iff: $\boldsymbol{u} \triangleright \boldsymbol{v}, \forall \boldsymbol{v} \in \boldsymbol{Q}$. 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:

$\boldsymbol{X_{true}} = \{\boldsymbol{x} \in \boldsymbol{X_f} \mid \boldsymbol{x}$ 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. 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. We assume a network with a reservation model and QoS guarantees.
Let be:

- $(i,j) \in E$: Link from node i to node j; $i, j \in V$.
- $c_{ij} \in \Re^+$: Cost per bps of link (i,j).
- $d_{ij} \in \Re^+$: Propagation Delay of link (i,j). Queuing delay is not considered for this model.
- $z_{ij} \in \Re^+$: Capacity of link (i,j).
- $t_{ij} \in \Re^+$: Current traffic of link (i,j).
- $S \in V$: Source node of a multicast group.
- $N_r \subseteq V\text{-}\{s\}$: Set of destinations of a multicast group.
- $n_i \in N_r$: One of $|N_r|$ destinations, where $|.|$ indicates cardinality.
- $\phi \in \Re^+$: Traffic demand, in bps.
- $T(s,N_r)$: Multicast tree with source in s and set of destinations N_r.
- $p_T(s, n_i) \subseteq T(s,N_r)$: Path connecting source s to a destination $n_i \in N_r$. Note that $T(s,N_r)$ represent a solutions $\boldsymbol{x}$ in Section 3.
- $d(p_T(s, n_i))$: Delay of the path $p_T(s,n_i)$, given by:

$$d(p_T(s,n_i)) = \sum_{(i,j) \in p_T(s,n_i)} d_{ij} \qquad (1)$$

Using the above definitions, a multicast routing problem for Traffic Engineering may be stated as a MOP that tries to find the multicast tree $T(s,N_r)$ that simultaneously minimizes the following objective functions:

1- Maximum link utilization of the tree:

$$\alpha(T) = \underset{(i,j) \in T}{Max} \left\{ \frac{\phi + t_{ij}}{z_{ij}} \right\} \qquad (2)$$

2- Cost of the multicast tree:

$$C(T) = \phi . \sum_{(i,j) \in T} c_{ij} \qquad (3)$$

3- Maximum end-to-end delay of a multicast tree:

$$DM(T) = \underset{n_i \in N_r}{Max} \{d(p_T(s, n_i))\} \qquad (4)$$

4- Average delay of a multicast tree:

$$DA(T) = \frac{1}{N_r} \cdot \sum_{n_i \in N_r} d(p_T(s, n_i)) \qquad (5)$$

The problem is subject to a link capacity constraint:

$$\phi + t_{ij} \leq z_{ij} \qquad \forall (i, j) \in T(s, N_r) \qquad (6)$$

Notice that $\boldsymbol{x = T(s, N_r)}$ and $\boldsymbol{y = [\alpha(T)\ \ C(T)\ \ DM(T)\ \ DA(T)]}$. A simple example follows to clarify the notation defined above.

Example 1. Given the network topology of Figure 1 [7], the numbers over each link *(i,j)* denotes d_{ij} in *ms*, c_{ij} and t_{ij} at the current time (in Mbps). For each link, z_{ij}=1.5 Mbps. Let suppose a traffic request arriving with ϕ = 0.2 Mbps, *s*=5, and *N*={0, 2, 6, 13}. Figure 1 shows the multicast tree (*T*) constructed with MMA. For this work $T \equiv T(s,N_r)$ for further simplicity.

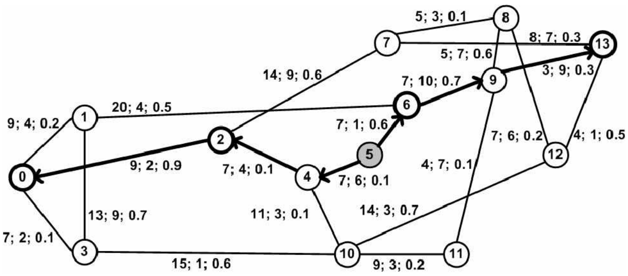


Figure 1: The NSF Net. d_{ij}, c_{ij} and t_{ij} are shown over each link. Objective Functions are $\alpha(T)$=0.73, $C(T)$=6.4, $DA(T)$=16.5, $DM(T)$=23. Multicast group: $s=5$ & $N_r=\{0,2,6,13\}$.

Table 1 presents the objective functions calculated for the solution of Figure 1.

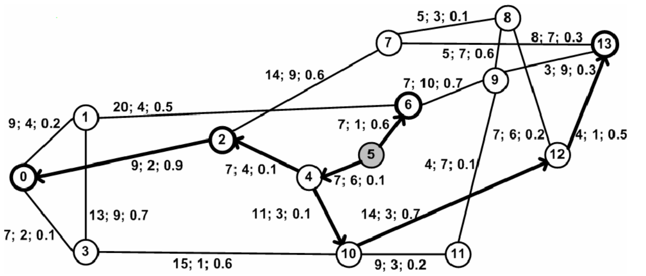
For the same example, Figure 2 presents in (a), (b) and (c) three different alternatives of solution trees, for the same multicast group, to clarify the concept of non-dominance. Notice that each tree is better than each other in at least one objective.

It is important to notice, from the mathematical formulation that the four objective functions are treated independently and should be minimized simultaneously. They are not combined to form a scalar single-objective function through a linear combination (as weighted sum) nor are any of them treated as a restriction. This way, using the concept of dominance, a whole set of optimal Pareto solutions is calculated in one run.

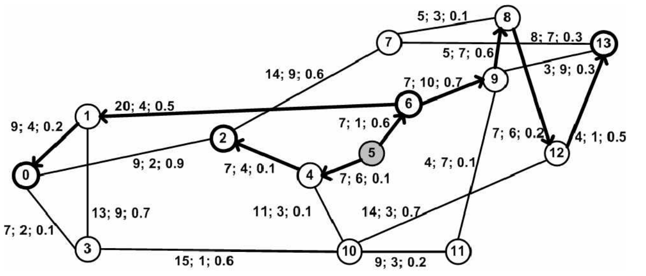
For the presented example the set of optimal Pareto set is shown in Table 2. The objectives functions are presented in Table 3. Notice that solution S_1 corresponds to Figure 2(a), S_2 corresponds to Figure 2(b) and S_3 corresponds to Figure 3(c).

Table 1: Objective Functions Calculated for Example 1

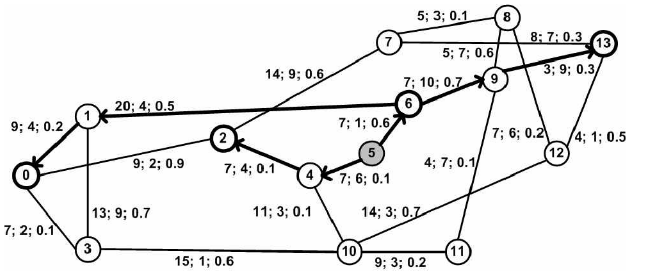
	Tree					
(i,j)	*(5,4)*	*(4,2)*	*(2,0)*	*(5,6)*	*(6,9)*	*(9,13)*
d_{ij}	7	7	9	7	7	8
c_{ij}	6	4	2	1	10	9
t_{ij}	0.1	0.1	0.9	0.6	0.7	0.8
$(\phi+t_{ij})/z_{ij}$	0. 2	0.2	***0.73***	0.53	0.6	0.53
$d(p_T(5,4))$	$d_{5,4} = 7$					
$d(p_T(5,2))$	$d_{5,4} + d_{4,2} = 7+7=14$					
$d(p_T(5,0))$	$d_{5,4} + d_{4,2} + d_{2,0} = 7+7+9=$***23***					
$d(p_T(5,6))$	$d_{5,6} = 7$					
$d(p_T(5,13))$	$d_{5,6} + d_{6,9} + d_{9,13} = 7+7+8=22$					
	Metrics of the solution Tree					
$\alpha(T)$	***0.73***					
C(T)	0.2*(6+4+2+1+10+9) = ***6.4***					
DA(T)	(7+14+16+7+22)/4 = ***16.5***					
DM(T)	***23***					



(a) $\alpha(T)$=0.73, $C(T)$=5.0, $DA(T)$=20.0, $DM(T)$=36.



(b) $\alpha(T)$=0.6, $C(T)$=8.6, $DA(T)$=21.75, $DM(T)$=36.0



(c) $\alpha(T)$=0.67, $C(T)$=7.6, $DA(T)$=19.75, $DM(T)$=36.0

Figure 2: The NSF Net. (a) to (c) shown different alternatives tress for the multicast group with *s*=5, N_r={0, 2, 6, 13} and ϕ=0.2 Mbps.

Table 2: Optimal Pareto set for Example 1

	Tree
S_1	(5,4),(5,6),(4,2),(4,10),(2,0),(10,12),(12,13)
S_2	(5,6),(5,4),(6,1),(6,9),(4,2),(1,0),(9,8),(8,12),(12,13)
S_3	(5,6),(5,4),(6,1),(6,9),(4,2),(1,0),(9,13)
S_4	(5,6),(5,4),(6,9),(4,2),(9,13),(2,0)
S_5	(5,6),(5,4),(6,1),(4,2),(4,10),(1,0),(10,12),(12,13)
S_6	(5,6),(5,4),(6,1),(4,2),(4,10),(1,0),(10,12),(12,13)
S_7	(5,6),(6,1),(1,0),(0,3),(0,2),(3,10),(10,12),(12,13)
S_8	(5,6),(5,4),(6,1),(4,2),(1,0),(2,7),(7,13)
S_9	(5,6),(5,4),(4,2),(4,10),(10,12),(10,3),(12,13),(3,0)
S_{10}	(5,6),(5,4),(4,10),(10,3),(10,12),(3,0),(12,13),(0,2)

Table 3: Objectives Vectors

	$\alpha(T)$	$C(T)$	$DA(T)$	$DM(T)$
S_1	0.73	5	20	36
S_2	0.6	8.6	21.75	36
S_3	0.67	7.6	19.75	36
S_4	0.73	6.4	16.50	23
S_5	0.73	4	26.75	63
S_6	0.6	6.2	23.25	36
S_7	0.73	3.6	41	76
S_8	0.53	7	23.75	38
S_9	0.6	5.2	24.25	4
S_{10}	0.73	4.8	33	49

5. ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) is a metaheuristic inspired by the behavior of ant colonies [15]. In the last few years, ACO has received increased attention by the scientific community as can be seen by the growing number of publications and the different fields of application [5]. Even though, there are several ACO variants, what can be considered a standard approach is next presented [16].

ACO is especially appealing when constructing solutions are needed, therefore, it seem interesting to study its application to the Multicast Problem.

Standard Approach. ACO uses simple agents called *ants* and a pheromone matrix $\tau=\{\tau_{ij}\}$ for constructing iteratively candidate solutions. The initial values is $\tau_{ij}=\tau_0$ $\forall(i,j)\in E$, where $\tau_0>0$. Furthermore, it takes advantage of heuristic information using a parameter $\eta_{ij}=1/d_{ij}$ called *visibility*. The relative influence between the heuristic information and the pheromone levels are define for parameters α and β. While an ant is visiting node i, N_i represents the set of neighbor nodes that are not yet visited. The probability of choosing a node j while at node i, is defined in the equation (7).

$$p_{ij} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}.\eta_{ij}^{\beta}}{\sum_{\forall g\in N_i}\tau_{ig}^{\alpha}.\eta_{ig}^{\beta}} & \text{if } j \in N_i \\ 0 & \text{otherwise} \end{cases} \qquad (7)$$

At every generation of the algorithm, each ant of a colony constructs a complete solution T using (7), starting at source node s. Pheromones evaporation is applied for all (i,j) according to $\tau_{ij}=(1-\rho).\tau_{ij}$, where parameter $\rho\in(0;1]$ determines the evaporation rate. Considering an elitist strategy, the best solutions found so far T_{best} updates τ according to $\tau_{ij}=\tau_{ij}+\Delta\tau$, where $\Delta\tau(i,j)=1/f(T_{best})$ if $(i,j)\in T_{best}$ and $\Delta\tau(i,j)=0$ if $(i,j)\notin T_{best}$.

Note that the standard approach optimizes a single objective. In the next Sections this work presents a multiobjective approach based on the presented standard ACO.

6. MULTIOBJECTIVE ANT COLONY OPTIMIZATION

Following *the Multi-Objective Ant Colony Optimization Algorithm* (MOACS) scheme [4], which is a generalization of the ACS [17], the proposed algorithm uses a colony of ants (or agents) and pheromone matrix $\tau=\{\tau_{ij}\}$ for the construction of w solutions T at every generation. This new approach also takes advantage of three heuristics information of the multicast routing problem, using $\eta_{dij}=1/d_{ij}$, $\eta_{cij}=1/c_{ij}$ and $\eta_{tij}=1/t_{ij}$. Parameters α and β define the relative influence between the heuristics information and the pheromone levels.

Additionally, this work also proposes variables λ^d, λ^c and λ^t, which define the relative influence among heuristics information. Then, a known Pareto Front Y_{known} [14] is updated including the best non-dominated solutions that have been calculated so far. Finally, the gathered information is saved updating a pheromone matrix τ_{ij}. Figure 2 presents the general procedure of the proposed MOACS. In general, if the state of Y_{known} was changed, the pheromone matrix τ_{ij} is re-initialized ($\tau_{ij}=\tau_0$ $\forall(i,j)\in E$) to improve exploration in the decision space $\boldsymbol{X}$. Otherwise, τ_{ij} is globally updated using the solutions of Y_{known} to better exploit the knowledge of the best known solutions. Note that only the links of found solutions T in Y_{known} are used to update the pheromone matrix τ_{ij}.

```
begin MOACS
  Read group (s, N_r), traffic demands φ, table t_ij, α, β and ρ
  Initialize τ_ij with τ_0      /*τ_0 is the initial level of τ_ij
  do {
     for λ^d = 0 to m-1
        for λ^c = 0 to m-1
           λ^t = m-1- λ^c
           T = Build Tree (α, β, ρ, λ^d, λ^c, λ^t, φ, (s, N_r), t_ij)
           if (T is not dominated by any T_x ∈ Y_known) then
              Y_known = Y_known ∪ T – {T_y | T ≻ T_y},∀T_y ∈ Y_known
           end if
        end for
     end for
     if (Y_known was modified) then
        τ_ij = τ_0 ∀(i,j) ∈ E
     else
        repeat ∀T_k ∈ Y_known
          τ_ij = (1-ρ).τ_ij + ρ.Δτ^k  ∀(i,j) ∈ T_k
        end repeat
      end if
  } while stop criterion is not verified
  Return Y_known
end MOACS
```

Figure 2: General Procedure of MOACS.

where:

$$\Delta\tau^{k} = \frac{1}{\alpha(T_k)+C(T_k)+DA(T_k)+DM(T_k)} \tag{8}$$

with:

$\alpha(T_k)$ -normalized maximum link utilization, given by (2)
$C(T_k)$ -normalized solution cost, given by (3)
$DA(T_k)$ -normalized maximum end-to-end delay, given by (4)
$DM(T_k)$ -normalized average delay, given by (5)
$\rho \in (0, 1]$ - trail persistence.
For normalization purposes, each objective function is divided by and *a priori* maximum value.

To construct a solution, an ant begins its job in the source node *s*. A non-visited node is pseudo-randomly selected at each step [4]. The pseudo-random procedure is presented in Figure 3, while equation (9) gives the probability to select a link. This process continues until all the destination nodes of the multicast group are reached. Considering *R* as the list of starting nodes, N_i as the list of feasible neighboring nodes to the node *i*, D_r as the set of destination nodes already reached. Procedure to find a solution *T* is summarized in Figure 4.

```
begin
  Select randomly q          /* q, q0 ∈ (0,1]
  if q > q0 then
    Choose node j with larger pij
  else
    Randomly choose j using probability pij
  end if
end
```

Figure 3: Pseudo-random Rule for selecting a node *j* of N_i.

The following is the probability assigned to link *(i,j)* with three heuristic visibilities:

$$p_{ij} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\cdot\left([\eta_{dij}]^{\lambda^d}\cdot[\eta_{cij}]^{\lambda^c}\cdot[\eta_{tij}]^{\lambda^t}\right)^{\beta}}{\sum_{\forall g\in N_i}\tau_{ig}^{\alpha}\cdot\left([\eta_{dig}]^{\lambda^d}\cdot[\eta_{cig}]^{\lambda^c}\cdot[\eta_{tig}]^{\lambda^t}\right)^{\beta}} & \text{if } j\in N_i \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

```
begin Build Tree
  Read α, β, ρ, λd, λc, λt, φ, (s, Nr), tij
  T = ∅; Dr = ∅; R = s
  do {
    Select node i of R and build set Ni
    if (Ni = ∅) then
      R = R – i;          /* erase node without feasible neighbor
    else
      Assign probability pij to each node of Ni
      Select node j of Ni using Pseudo-random Rule
      T = T ∪ (i, j);   R = R ∪ j;
      if (j ∈ Nr) then
        Dr = Dr ∪ j       /* node j is a destination node
      end if
    end if
    τij = (1 - ρ).τij + ρ.τ0   /* update pheromone
  } while (R ≠ ∅ or Dr ≠ Nr)
  Prune Tree T             /* eliminate not used links
  Return T
end Build Tree
```

Figure 4: Procedure to Build Tree.

7. MULTIOBJECTIVE MAX-MIN ANT SYSTEM.

The standard *Max-Min Ant System* (MMAS) presented by Stützle and Hoos [5], was derived from the standard ACO [16] and it incorporated three key features to achieve a better performance:

- Only the best solution at each iteration or during the execution of the algorithm updates the pheromone trail τ.
- A range $[\tau_{min},\tau_{max}]$ is imposed to τ components. The upper level may be calculated as $\tau_{max}=1/(f(T).(1-\rho))$ while the lower level as $\tau_{min}=\tau_{max}/2.w$, where *w* is the number of ants at each generation.
- Initialize the pheromone trails with τ_{max}, achieving a high exploration at the start of the algorithm.

Given that MMAS is mono-objective; this work modifies MMAS to solve multiobjective problems, with the following changes:

- The new Multiobjective MMAS (or more simply, M-MMAS) finds a whole set of Pareto optimal solutions called Y_{known} instead of finding a single optimal solution.
- To guide the ants in the search space, three heuristics information are proposed: $\eta_{dij}=1/d_{ij}$, $\eta_{cij}=1/c_{ij}$ and $\eta_{tij}=1/t_{ij}$. Variables λ^d, λ^c and λ^t determine the relative influence among heuristics information. Therefore, the probability of choosing a node *j* while an ant visits node *i*, is given by equation (9).
- The pheromone matrix τ is updated according to $\tau_{ij}=\tau_{ij}+\Delta\tau^k$ $\forall(i,j)\in T_k$ and $\forall T_k\in Y_{known}$ up to an upper level $\tau_{max}=\Delta\tau^k/(1-\rho)$ and not below a minimum level $\tau_{min}=\Delta\tau^k/2w(1-\rho)$. The pheromone level $\Delta\tau^k$, is given by equation (8).

Figure 5 presents the general M-MMAS procedure. The algorithm builds a solution tree *T* using the same general ideas used for the MOACS but with two small differences:

(a) *the pheromone update,* and
(b) it does not use *pseudo-random rule* (given in Figure 3).

```
begin M-MMAS
 Read group (s, Nr), traffic demands φ, table tij, α, β and ρ
 Initialize τij with τmax      /*τmax is the initial level of τij
 do {
   for λd = 0 to m-1
     for λc = 0 to m-1
       λt = m-1- λc     /*Note that w=m.m
       T = Build Tree (α, β, ρ, λd, λc, λt, φ, (s, Nr), tij)
       if (T is not dominated by Tx ∈ Yknown) then
         Yknown = Yknown ∪ T – {Ty ∈ Yknown | T ≻ Ty}
       end if
     end for
   end for
   τij = (1-ρ).τij        ∀(i,j)∈E
   if τij < τmin then τij = τmin ∀(i,j)∈E
   repeat ∀Tk∈ Yknown
     τij = τij + Δτk    ∀(i,j) ∈ Tk
     if τij > τmax then  τij = τmax  ∀(i,j)∈ Tk
   end repeat
 } while stop criterion is not verified
 Return Yknown
end M-MMAS
```

Figure 5: General Procedure of the Multiobjective MMAS.

The election of links is carried out randomly with probability p_{ij} given by equation (9). Initially, M-MMAS reads the parameters and initializes the pheromone matrix τ. At each generation, w solutions T are built. The set Y_{known} is updated with non-dominates solutions T while dominated solutions of Y_{known} are eliminated. To update pheromone matrix τ, evaporation is first performed and pheromone is latter added $\forall (i,j) \in T_k$ and $\forall T_k \in Y_{known}$.

8. MULTIOBJECTIVE MULTICAST ALGORITHM

Multiobjective Multicast Algorithm (MMA), recently proposed in [7, 8, 9], is based on the *Strength Pareto Evolutionary Algorithm* (SPEA) [10]. MMA holds an evolutionary population P and an external Pareto solution set P_{nd}. Starting with a random population P of solutions, the individuals evolve to Pareto optimal solutions to be included in P_{nd}. A general MMA procedure is shown in Figure 6, while its codification is represented in Figure 7.

MMA evolutionary algorithm begins reading the variables of the problem and basically proceeds as follows (see pseudo-code in Figure 6):

Build routing tables: For each $n_i \in N_r$, a routing table is built. It consists of the R shortest, R cheapest and R least used paths. R is a parameter of the algorithm. A chromosome is represented by a string of length $|N_r|$ in which each element (gene) g_i represents a path between source s and destination n_i. See Figure 7 to see a chromosome that represents the tree in Figure 7.

```
begin MMA
  Read (s,N_r) , t_ij and φ
  Build routing tables
  Initialize P
  do {
    Discard individuals
    Evaluate individuals
    Update non-dominated set P_nd
    Compute fitness
    Selection
    Crossover and mutation
  } while stop criterion is not verified
end MMA
```

Figure 6: Procedure General of MMA.

Discard individuals: In P, there may be duplicated chromosomes.

Thus, new randomly generated individuals replace duplicated chromosomes.

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

Compute fitness: Fitness is computed for each individual, using SPEA procedure [10].

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

Crossover and Mutation: MMA uses two-point crossover operator over selected pair of individuals. Then, some genes in each chromosome of the new population are randomly changed (mutated), obtaining a new solution. The process continues until a stop criterion, as a maximum number of generations, is satisfied.

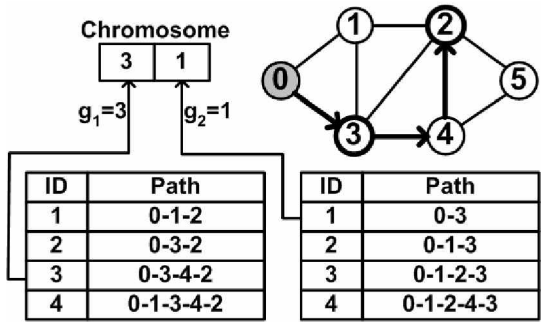


Figure 7: Relationship between a chromosome, genes and routing tables for a tree with $s=0$ and $N_r=\{2, 3\}$.

9. EXPERIMENTAL ENVIRONMENT

Simulations were carried out using the NTT network topology illustrated in Figure 8 [7]. We have performed many simulations with many multicast groups along our simulations [18]. But, we have chosen only two of them for brevity reasons. So, in Table 4 we show two multicast groups that were used for the experiments that follow. For each group, experimental results are analyzed after 160 and 320 seconds. Initially, the network was considered 50% randomly loaded on average, i.e. the initial traffic t_{ij} is around 50% of its total load capacity z_{ij}.

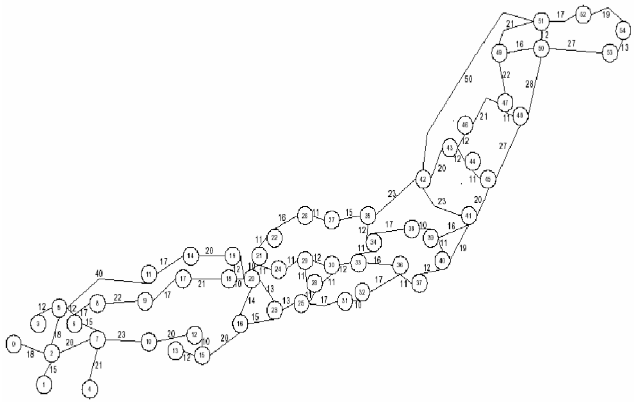


Figure 8: Japan NTT network with 55 nodes and 144 links used for the simulations. Over each link (i,j), a delay d_{ij} is shown.

Three algorithms (*MOACS*, *M-MMAS* & *MMA)* have been implemented on a 350 MHz AMDK6 computer with 128 MB of RAM. A Borland C++ V 5.02 compiler was used. For these experiments, the results of the proposed *MOACS* and *M-MMAS* were compared to the evolutionary algorithm *MMA* [7, 8, 9]. Experimental results are summarized in Section 10.

Table 4: Multicast Group used for the tests. Each Group has one source and | N_r| destinations

Test Group	*Source {s}*	*Destinations {N_r}*	$\|N_r\|$
Group 1 (*small*)	5	{0,1,8,10,22,32,38,43,53}	9
Group 2 (*large*)	4	{0,1,3,5,6,9,10,11,12,17,19,21,22, 23,25,33,34,37,41,44,46,47,52,54}	24

To calculate an approximation to the true Pareto Front, Y_{apr}, the following six-step procedure was used:

1. Each algorithm (*MOACS, M-MMAS & MMA*) was run five times and an average was calculated for comparison to each other.

2. For each algorithm, five sets of non-dominated solutions: Y_1, $Y_2 \dots Y_5$, were calculated, one for each run.

3. For each algorithm, overpopulation Y_T was obtained, where

$$Y_T = \bigcup_{i=1}^{5} Y_i \ .$$

4. Dominated solutions were deleted from Y_T, obtaining the Pareto Front calculated by each algorithm, as follows:

Y_{MOACS} (Pareto Front obtained with five runs, using *MOACS),*

Y_{M-MMAS} (Pareto Front obtained with five runs, using *M-MMAS)*

Y_{MMA} (Pareto Front obtained with five runs, using *MMA*).

5. A set of solutions *Y'* was obtained as $Y' = Y_{MOACS} \cup Y_{M-MAS} \cup Y_{MMA}$.

6. Dominated solutions were deleted from *Y'*, and an approximation of Y_{true}, called Y_{apr}, is finally created. Note that for practical issues $Y_{apr} \approx Y_{true}$, i.e. Y_{apr} is an excellent approximation of Y_{true}.

Table 5 presents the total number of solutions $|Y_{apr}|$ that were experimentally found for each multicast group.

Table 5: Total number of non-dominated solutions belonging to Y_{apr} for each multicast group

	Group 1 (*small*)	Group 2 (*large*)
$\|Y_{apr}\|$	30	56

10. EXPERIMENTAL RESULTS

The following tables show a comparison between the solutions found with the implemented algorithms (*MOACS, M-MMAS & MMA*) with respect to Y_{apr}. At the same time, algorithms are compared using the *coverage* figure of merit that counts the average number of solutions dominated by the other algorithm's Pareto set [10], as shown in Tables 6 to 9. To understand those tables, the following notation is used:

- $\in Y_{apr}$ average number of solutions that are in Y_{apr};
- $Y_{apr}\succ$ average number of solutions that are dominated by Y_{apr};
- $|Y_{alg}|$ average number of solutions found by each algorithm;
- $\%Y_{apr}$ percentage of solutions found by a given algorithm, i.e. 100. ($\in Y_{apr}$) / ($|Y_{apr}|$).

10.1 Results obtained for Multicast Group 1

Tables 6 and 7 present experimental results obtained for the small-multicast group 1 (see Table 5) after a run of 160 seconds and 320 seconds respectively. Both Tables show that *MOACS* and *M-MMAS* found a lot more solutions than *MMA*. Considering only both ACOs, *MOACS* found more solutions $\in Y_{apr}$, giving a better approximation to the Pareto front.

Table 6: Small Multicast Group 1 – Run time = 160 seconds

Comparison of Solutions with Y_{apr}					Covering among Algorithms		
	$\in Y_{apr}$	$Y_{apr}\succ$	$\|Y_{alg}\|$	$\%Y_{apr}$	Y_{MOACS}	Y_{MMMAS}	Y_{MMA}
Y_{MOACS}	29	1	30	97%		2.5	1
Y_{MMMAS}	3.8	8.6	12.4	13%	0		1
Y_{MMA}	3.4	2.6	6	11%	1	1	

Table 7: Small Multicast Group 1 – Run time = 320 seconds

Comparison of Solutions with Y_{apr}					Covering among Algorithms		
	$\in Y_{apr}$	$Y_{apr}\succ$	$\|Y_{alg}\|$	$\%Y_{apr}$	Y_{MOACS}	Y_{MMMAS}	Y_{MMA}
Y_{MOACS}	29	1	30	97%		2	2
Y_{MMMAS}	7	8.6	15.6	23%	0		1
Y_{MMA}	4.4	3	7.4	15%	1	2	

10.2 Results obtained for Multicast Group 2

Tables 8 and 9 present experimental results obtained for (a large) multicast Group 2 after a run of 160 seconds and 320 seconds respectively. Both tables show that *MOACS* found more solutions than *M-MMAS* and *MMA*. However, when coverage is considered *MMA* dominates more solutions of *MOACS* and it dominates more solutions of *M-MMAS*, proving its ability to find very good solutions even though, it did not find a large number of solutions. Once more, *MOACS* found the best approximation to the Pareto front Y_{apr}.

Table 8: Large Multicast Group 2 – Run time = 160 seconds

Comparison of Solutions with Y_{apr}					Covering among Algorithms		
	$\in Y_{apr}$	$Y_{apr}\succ$	$\|Y_{alg}\|$	$\%Y_{apr}$	Y_{MOACS}	Y_{MMMAS}	Y_{MMA}
Y_{MOACS}	16	17.6	33.6	29%		3	1.5
Y_{MMMAS}	4.4	8.6	13	37%	2		2
Y_{MMA}	6.2	4.8	11	8%	4.2	2	

Table 9: Large Multicast Group 2 – Run time = 320 seconds

Comparison of Solutions with Y_{apr}					Covering among Algorithms		
	$\in Y_{apr}$	$Y_{apr}\succ$	$\|Y_{alg}\|$	$\%Y_{apr}$	Y_{MOACS}	Y_{MMMAS}	Y_{MMA}
Y_{MOACS}	22	14.4	36.4	39%		2	2
Y_{MMMAS}	6.6	11.2	17.8	12%	0		1
Y_{MMA}	2.2	1.2	3.4	4%	4.2	1	

10.3 General Average

Table 10 presents general averages of the comparison metrics already defined, considering all performed experiments. It can be noticed that, on average, *MOACS* is superior to *M-MMAS* and *MMA*. In fact, *MOACS* found in average 65.5% of Y_{apr} solutions, while *M-MMAS* and *MMA* just found 13.5% and 10.3%

respectively. Also considering Coverage, *MOACS* looks better given that it dominates more solutions calculated by *M-MMAS* and *MMA*. Finally, it should be mentioned that *MMA* presented a slightly better performance than *M-MMAS* given that it dominates more solutions of *M-MMAS*.

Table 10: General averages of comparison figures of merit

Comparison of Solutions with Y_{apr}					Covering among Algorithms		
	$\in Y_{apr}$	$Y_{apr\succ}$	$\lvert Y_{alg}\rvert$	$\%Y_{apr}$	Y_{MOACS}	Y_{MMMAS}	Y_{MMA}
Y_{MOACS}	24	8.5	32.5	65.5%		2.3	1.6
Y_{MMMAS}	4.7	9.25	13	13.5%	0.5		1.25
Y_{MMA}	4.5	2.9	6.9	10.3%	1.5	1.5	

11. CONCLUSIONS

This paper introduces a new approach based on *MOACS* and *MMAS* to solve the multicast routing problem. *MOACS* and Multiobjective *MMAS* are able to optimize simultaneously four objective functions, such as: (1) maximum link utilization, (2) cost of a routing tree, (3) maximum end-to-end delay and (4) average delay. These new proposals are able to solve a multicast routing problem in a truly multiobjective context, considering all four objectives at the same time, for the first time using an algorithm based on Ant Colony Optimization. The new approaches calculate not only one possible solution, but also a whole set of optimal Pareto solutions in only one run. This last feature is especially important since the most adequate solution can be chosen for each particular case without *a priori* restrictions that may eliminate good solutions.

To validate the new approaches, *MOACS* and *M-MMAS* were compared to the *MMA*, a representative algorithm for solving the considered multicast routing problem in a truly multiobjective context, for Traffic Engineering. The experimental results showed that *MOACS* and *M-MMAS* are able to find more solutions than *MMA* for different running time and various multicast groups. Furthermore, *MOACS* solutions covered *M-MMAS* and *MMA* solutions most of the time, i.e. *MOACS* found better solutions in average than *M-MMAS* and *MMA*. Therefore, *MOACS* approach is the one recommended, considering the presented experimental results.

The main contribution of this paper is the resolution of the multiobjective multicast routing problem for the first time in the literature, using an ACO algorithm. With this aim, the MMAS traditional one-objective algorithm is modified to solve a multiobjective problem.

As a future work, the authors will perform more tests over other network topologies and using other multiobjective metrics.

12. REFERENCES

[1] A. Tanenbaum, *Computer Networks*, Prentice Hall 4º Edition, 2003.

[2] Y. Seok, Y. Lee, Y. Choi, and C. Kim, "Explicit multicast routing algorithm for constrained traffic engineering", *IEEE 7th International Symposium on Computer and Communications* (ISCC'02). Italy, 2002.

[3] Y. Donoso, R. Fabregat, and J. Marzo, "Multiobjective optimization algorithm for multicast routing with traffic engineering", *IEEE 3rd International Conference on Networking* (ICN'2004), Guadalupe, French Caribbean, March – 2004.

[4] M. Schaerer, and B. Barán. **"**A Multiobjective Ant Colony System For Vehicle Routing Problem With Time Windows**",** *IASTED International Conference on Applied Informatics*, Innsbruck, Austria, 2003.

[5] T. Stützle and H. Hoos. "MAX-MIN Ant System". *Future Generation Computer System*. June 2000.

[6] F. Talavera and B. Barán. "Policies for Dynamical Muti-Objective Environment of Multicast Traffic Enginering". 12th International Conference on Telecommunications. Cape Town, Sudáfrica, 2005.

[7] J. Crichigno and B. Barán. "Multiobjective Multicast Routing Algorithm for Traffic Engineering". *IEEE International Conference on Computer and Communications* ICCCN'2004, Chicago, US, 2004.

[8] J. Crichigno and B. Barán. "Multiobjective Multicast Routing Algorithm". *IEEE International Conference on Telecommunications* - ICT'2004, Ceara, Brazil, 2004.

[9] J. Crichigno and B. Barán, "A Multicast Routing Algorithm Using Multiobjective Optimization". *IEEE International Conference on Telecommunications* - ICT'2004, Ceara, Brazil, 2004.

[10] E. Zitzler, and L. Thiele, "Multiobjective Evolutionary Algorithms: A comparative Case Study and the Strength Pareto Approach", *IEEE Transactions on Evolutionary Computation*, Volume 3, No. 4, 1999, pp 257-271.

[11] Y. Liu, and J. Wu. "The degree-constrained multicasting algorithm using ant algorithm" *IEEE 10th International Conference on Telecommunications*" 2003.

[12] J. Gu, C. Chu, X. Hou, and Q. Gu. "A heuristic ant algorithm for solving QoS multicast routing problem" *IEEE Transactions on Evolutionary Computation*, 2002. CEC '02. Volume 2, pp 1630-1635.

[13] M. Dorigo, and L. M. Gambardella. "Ant Colony System: A cooperative learning approach to the traveling salesman problem" *IEEE Transactions on Evolutionary Computation,* 1: 1, pp 53-66, 1997.

[14] D. A. Van Veldhuizen. "Classifications, Analyses and New Innovations" *Ph. D. thesis.* Air Force Institute of Technology, 1999.

[15] M. Dorigo and G. Di Caro. "The Ant Colony Optimization meta-heuristic". In *New Ideas in Optimization*. McGraw Hill, London, UK, 1999.

[16] M. Guntsch and M. Middendorf. "*A Population Based Approach for ACO*". In Stefano Cagnoni, Jens Gottlieb, Emma Hart, Martin Middendorf, and Günther Raidl, *Applications of Evolutionary Computing, Proceedings of EvoWorkshops2002: EvoCOP, EvoIASP, EvoSTim,* Springer-Verlag ,vol. 2279, pages 71–80, Kinsale, Ireland, 2002.

[17] M. Dorigo, and L. M. Gambardella. “Ant Colony System: A cooperative learning approach to the traveling salesman problem”. *IEEE Transactions on Evolutionary Computation*, 1:1, pp 53-66, 1997.

[18] D. Pinto. “Enrutamiento Multicast Multiobjetivo basado en Colonia de Hormigas”. Final work - Electronic Engineering, Catholic University of Asunción, 2005.