Polices for Dynamical MultiObjective Environment of Multicast Traffic Engineering.

Francisco Talavera, Jorge Crichigno, Benjamín Barán, *Member IEEE*.

Abstract— **Recognized the multiobjective nature of multicast traffic engineering, this paper compares several Multiobjective Evolutionary Algorithms (MOEAs) to solve that problem and proposes several policies to choose a good solution from a Pareto set in a dynamical environment. Experimental results show that most MOEAs may suit for the task. Moreover, the chosen policy is the main factor to define performance in a dynamical environment. Therefore, seven different policies are proposed and tested in a dynamical environment, proving that the policy of choosing the closest solution to the origin provides excellent trade-off values.**

***Index Terms*—Multicast, Multiobjective Evolutionary Algorithms, Pareto Dominance, Traffic Engineering.**

I. 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, video on-demand (VoD) and e-learning. For these applications, the delay from a source to each destination becomes a variable of vital importance in audio and/or video multicast transmissions [2].

For traffic engineering, other important objectives are taken into account in the optimization of multicast routing algorithm as maximum link utilization and the "cost" of the tree, being understood by "cost" other metrics to be minimized like: hop count, total bandwidth consumption, etc. Therefore, the multicast traffic engineering problem has been recently recognized as a multi-objective optimization problem [1]-[10].

Given that multicast traffic engineering (MTE) may be treated as a multi-objective problem (MOP), it is worth mention that Multi-Objectives Evolutionary Algorithms (MOEAs) are already recognized as well suited to solve that kind of problems [11], [12]. In fact, each reviewed paper that recognizes MTE as a MOP [1]-[9] proposes some kind of MOEA, finding a complete set of Pareto solutions in a single execution [11].

F. T. Francisco Talavera. Universidad Católica Nuestra Señora de la Asunción and Universidad Nacional de Asunción (ftalavera@cnc.una.py).

J. C. Jorge Crichigno. Universidad Católica Nuestra Señora de la Asunción (jcrichigno@cnc.una.py).

B. B. Benjamín Barán Universidad Católica Nuestra Señora de la Asunción and Universidad Nacional de Asunción (bbaran@cnc.una.py).

As already stated, MTE problem has been recently studied as a MOP in several publications. Thus, routing algorithms for multiobjective multicast are presented in [1], [7], [8], considering different sets of objective functions as: tree cost, maximum end-to-end delay, average delay and maximum link utilization. These works were later improved with another MOEA that uses a different codification [9]. An interesting approach considering multi-trees is presented in [10], while a first publication in a wireless environment was found in [4]. Quality of Service (QoS) considerations in a MOP context was considered in [5], [6]. These multiobjective proposals for multicast routing are generalized in a *GMM model for Dynamic Multicast Groups* in [2].

Given that different MOEAs were already used to solve the MTE problem, the present work first compares four of the most promising multiobjective evolutionary algorithm:

- SPEA (Strength Pareto Evolutionary Algorithm)[13];
- SPEA2 (SPEA, version 2) [14];
- NSGA2 (Nondominated Sort Genetic Algorithm 2)[15];
- cNSGA2 (Controlled NSGA2)[16];

Later, policies for the selection of good solutions of a Pareto set are proposed for a dynamical environment.

The rest of the document is organized as follows: Section II presents the mathematical formulation of the problem and a brief introduction to evolutionary algorithms (EAs), as well as the codification used in this work. Section III summarizes an algorithm for generation of testing scenarios. The proposed policies for selecting solutions are presented in Section IV. Experimental results are discussed in Section V, while the final conclusions and future works are left for Section VI.

II. Problem Formulation

A. Mathematic model

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 and z_{ij}, c_{ij}, d_{ij} and $t_{ij} \in \Re^+$ be its capacity, cost per bps, delay and current traffic, respectively. Let's denote $s \in V$ as a source, $N \subseteq V - \{s\}$ as the set of destinations, and $\phi \in \Re^+$ as the traffic demand (in bps) of a 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 $n \in N$. Finally, let $d(p_T(s, n))$ represent the delay of the path $p_T(s, n)$ given by the sum of the link delays that conform the path, i.e.,

$$d\left(p_T(s,n)\right) = \sum_{(i,j)\in p_T(s,n)} d_{ij} \quad (1)$$

Using the above definitions, a multicast routing problem for traffic engineering may be stated as a MOP that tries to find the multicast tree $T(s,N)$ that simultaneously minimize the following objectives:

1- Maximum link utilization of the tree:

$$\alpha_T = \underset{(i,j)\in T}{Max} \left\{ (\phi + t_{ij}) / z_{ij} \right\}. \quad (2)$$

2- Cost of the tree:

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

3- Average delay:

$$D_A = \frac{1}{|N|} \sum_{n\in N} d\left(p_T(s,n)\right). \quad (4)$$

subject to link capacity constraint:

$$\phi + t_{ij} \le z_{ij} \ ; \ \forall (i,j) \in T(s,N), \quad (5)$$

where | | denotes cardinality.

B. *Proposed multiobjective evolutionary algorithms*

The four MOEAs were implemented as they were proposed in [13]-[16] and following the scheme proposed in [9]. They begin with a set of random configurations called initial population P. Each individual T_p in the population represents a potential solution of the problem. At each generation, the individuals are evaluated using an adaptability function. Based on this value, some individuals, called parents, are selected. The probability of selection of an individual is related to its adaptability. Then, a number of genetic probabilistic operators are applied to the parents to produce new individuals that will be part of a new population. The process continues until a stop criterion is satisfied.

III. Dynamic Testing Scenarios

Several MOEAs were already compared in a static environment [3]; however, the present work presents an empirical comparison of MOEAs and selection policies in dynamic scenarios. The network used in the simulations was the NTT-net [14], which consists of 55 nodes and 144 links. The values of z_{ij}, c_{ij} and d_{ij} were taken from [1]. Under these conditions, Ψ traffic requests were generated, simulating a dynamic situation in which they arrive one after another. The demands, in bps, were set between a minimum ϕ_{min} and a maximum ϕ_{max}. Similarly, the sizes of the groups were set between $|N|_{min}$ and $|N|_{max}$. The algorithm to generate multicast groups is shown in Fig. 1.

```
for i = 1 to Ψ
    group(i)  =  groupGenerator(|N|min, |N|max);
    Tini(i)   =  random(unif, 0,2000);
    Tfin(i)   =  Tini(i) + random(exp, 0, 60);
    ϕ(i)      =  random(unif, ϕmin, ϕmax);
end for
```

Fig. 1. Algorithm to create Ψ random multicast groups.

The function *groupGenerator* 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, the traffic demand is set to a value between ϕ_{min} and ϕ_{max}.

The following two figures were defined, in order to have an idea of the traffic in the network (a) *Accumulated traffic demand*:

$$\phi_A(t) = \sum_{i=1}^{M(t)} \phi_i \ , \quad (6)$$

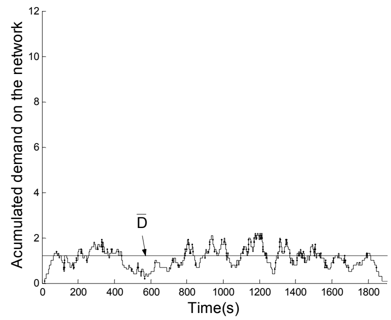
and (b) its *average over the simulated time*, denoted as $\overline{\phi}$, where $M(t)$ is the number of multicast group in the network at time t and ϕ_i the traffic demand of each multicast group i.

In order to compare the behavior of different MOEAs under several traffic loads over the network, three scenarios were defined: (a) low load, (b) high load and (c) saturation. Table I summarizes the parameter values used to generate each scenario.

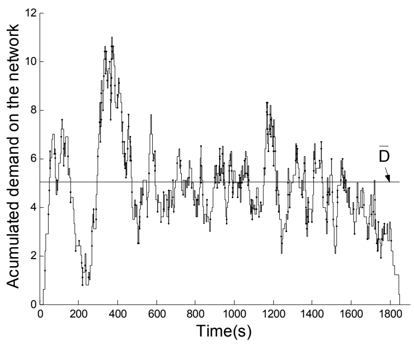
Table I. Parameters used in each scenario.

Scenarios	*Parameters*					
	$\overline{\phi}$	Ψ	$\lvert N\rvert_{min}$	$\lvert N\rvert_{max}$	ϕ_{min}	ϕ_{max}
Low load	1.208	200	4	10	0,1	0,2
High load	5.050	300	10	25	0,2	0,8
In saturation	7.463	400	10	35	0,2	0,8

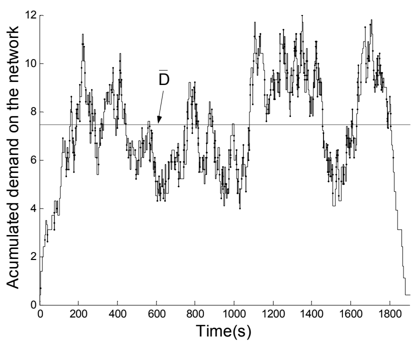
Fig. 2 shows the accumulated and average traffic demand of the three proposed scenarios. It can be appreciated that the low load scenario is pretty relaxed from the traffic point of view, while the high load scenario tests MOEAs under a much greater load than the previous scenario. In addition, each multicast group of the high load scenario consists of a greater number of destination nodes, which produces a higher bottleneck in the network. Finally, in the saturation scenario, the traffic demand is by far the greatest of the three scenarios. Moreover, the number of multicast group and the size of each one were set to a larger value than the previous ones to saturate the network with traffic, making too difficult for the network to route the whole traffic. Clearly, in the saturation scenario, some multicast groups are not routed, given the lack of network resources; therefore, the average number of rejected groups may be considered as an important metric to compare different algorithms and policies.



(a)



(b)



(c)

Fig. 2. Accumulated and average traffic demand of the three scenarios used in the simulations; (a) low load, (b) high load, and (c) saturation.

IV. Policies of Selection

From the fact that MOEAs provide a set of Pareto solutions, a mechanism is needed to choose a single alternative at each time a multicast group enters the network. Therefore, an interesting question arises. Which alternative of the Pareto set is the most suitable solution, being that all solutions are *non-comparable* to each other?. To find an answer, four selection policies are summarized in Table II.

A. Semi-static selection.

Considering the objective functions (α_T, C_T, D_A), this policy first defines an *acceptable upper bound* (α_{Tm}, C_{Tm}, D_{Am}). Table III summarizes the eight possible situations, indicating with a "1" that a given objective function exceeded the a priori upper bound, i.e., the value of that objective is not satisfactory. On the contrary, a "0" represents that an objective is acceptable, i.e., $\alpha_T \leq \alpha_{Tm}$, $C_T \leq C_{Tm}$ or $D_A \leq D_{Am}$. Clearly, at each decision time, those objectives with values larger than their upper bounds must have the highest priorities.

Table II. Summary of the proposed selection policies.

Policy	*Name*	*Symbol*	*Details*		
			Primary objective	Secondary objective	Third objective
Static*	S1	αC	α_T	C_T	D_A
	S2	$C\alpha$	C_T	α_T	D_A
	S3	αD_A	α_T	D_A	C_T
	S4	$D_A\alpha$	D_A	α_T	C_T
Semi-Static	SE	*SE1, SE2, SE3*	An acceptable upper bound (α_{Tm}, C_{Tm}, D_{Am}) is defined. With this parameter, the policy dynamically selects the most advisable solution, giving priorities to objectives that do not satisfy the desired operation point. Three different upper bounds were tested (*SE1, SE2* and *SE3*).		
Dynamic	Worse Acceptable Case	*DP*	A line between the origin of coordinates and the point (α_{Tm}, C_{Tm}, D_{Am}) is considered. The nearest solution to that line is chosen. That way, the 3 objectives may be simultaneously optimized, as shown in Fig. 3(a).		
	Origin of coordinates	*DC*	This dynamic policy selects the closest solution to the origin of coordinates, as shown in Fig. 3(b).		

*A static selection uses a "lexicographical order" [4].

The problem arises when there are more than two solutions with the same priority, i.e., there are two (or three) objectives that have not reached their upper limit or they have exceeded their upper bound. In these cases, it is needed a way to break the tie among objectives. Table III presents the criteria considered for this work. α_c, C_c and D_c are the time-depended values of the objective functions considering the whole network, as defined in equations (7) to (9).

Table III. Possible situations with a semi-static algorith.

Case	α_T	C_T	D_A	Criteria
1	0	0	0	$\alpha_c/\alpha_{Tm} = a$; $C_c/C_{Tm} = b$; $D_c/D_{Am} = c$. The greatest value among *a*, *b* and *c* defines the primary objective.
2	0	0	1	The greatest priority is assigned to the objective that does not satisfies its upper limit. A tie with other objectives is broken in the same way as in *case 1*.
3	0	1	0	
4	1	0	0	
5	0	1	1	A lower priority is assigned to the objective with "0," while a tie is broken using the same criterion as in *case 8*.
6	1	0	1	
7	1	1	0	
8	1	1	1	The higher value among (1-*a*), (1-*b*) and (1-*c*) defines the main objective (see *case 1*.)

$$D_c = \sum_{i=1}^{M(t)} D_A^i \Big/ |N|^i, \tag{7}$$

$$C_c = M(t)^{-1} \sum_{i=1}^{M} C_T^i, \tag{8}$$

$$\alpha_c = |V|^{-1} \sum_{(i,j)\in E} \alpha_{ij}, \tag{9}$$

where $M(t)$ denotes the number of multicast group at time t, D_A^i the average delay of group i, $|N|^i$ the number of destination nodes of group i, C_T^i the cost for group i, and $|V|$ the number of nodes in the network.

B. Dynamic selection DP

DP selects the nearest solution to the line defined by the origin of coordinates and the point (α_{Tm}, C_{Tm}, D_{Am}). That way, the three objectives are simultaneously optimized. The chosen point was (07, 13, 115). Fig. 3(a) shows an example with two objective functions.

C. Dynamic selection DC

DC chooses the nearest solution to the origin of coordinates. It is important to note that this policy does not have any a priori parameter. Fig. 3(b) shows an example with two objective functions.

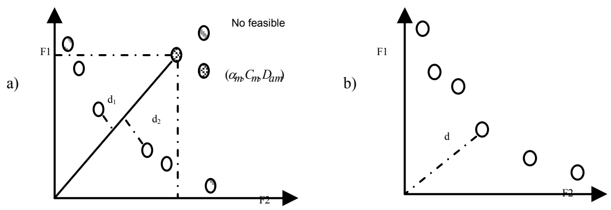


Fig. 3. Dynamic selection, (a) DP and (b) DC.

V. EXPERIMENTAL RESULTS.

For each scenario – low load, high load and saturation – ten runs were preformed and average values were calculated.

A. Comparison of MOEAs

The following performance figures were used to compare MOEAs:

Average maximum link utilization:

$$\alpha p = \frac{1}{M(t)} \sum_{w=1}^{M(t)} \alpha T(w) \quad (10)$$

Average cost of the trees:

$$Cp = \frac{1}{M(t)} \sum_{w=1}^{M(t)} CT(w) \quad (11)$$

Total average delay:

$$Dp = \frac{1}{H} \sum_{w=1}^{M(t)} \sum_{n \in N_w} d(p_T(s,n)) \quad (12)$$

where

$\alpha_T(w)$: maximum link utilization of group *w*

$C_T(w)$: cost of the tree of group *w*

H : total number of destination nodes.

After computing these metrics for each run, the average values in ten runs were calculated (denoted as $\overline{\alpha_p}$, $\overline{C_p}$ and $\overline{D_p}$ respectively.) Another metric is considered: *Na,* defined as the average number of traffic requests rejected for lack of network resources. Tables IV to VI show these metrics for each scenario. Note that the performance of each MOEA was quite similar to the other ones. It is important to highlight the difference between these results and the ones in [3] that only compares MOEAs in a static environment, using as performance metric the number of calculated Pareto solutions. Clearly, the fact that an algorithm finds more Pareto solutions does not imply a better performance in dynamic scenarios given that only one solution is chosen at a time. In summary, the selection of the MOEA to be used is not really relevant, given that any good implementation may give satisfactory results with a good selection policy.

The best solution for a certain column.
The second better solution for a certain column.

TABLE IV. RESULTS FOR SCENARIO 1: LOW LOAD.

	Averages			
	$\overline{\alpha_p}$	$\overline{C_p}$	$\overline{D_p}$	Na
SPEA2	0.11868	20.633	113.2032	0
SPEA	0.11882	20.633	113.249	0
CNSGA2	0.11855	20.633	113.1603	0
NSGA2	0.11855	20.633	113.1424	0

TABLE V. RESULTS FOR SCENARIO 2. HIGH LOAD.

	Averages			
	$\overline{\alpha_p}$	$\overline{C_p}$	$\overline{D_p}$	Na
SPEA2	0.57805	140.2427	115.2895	1
SPEA	0.57614	140.0172	115.7672	1
CNSGA2	0.57841	140.0833	115.531	1
NSGA2	0.57686	140.0906	115.5096	1

TABLE VI. RESULTS FOR SCENARIO 3. SATURATION.

	Averages			
	$\overline{\alpha_p}$	$\overline{C_p}$	$\overline{D_p}$	Na
SPEA2	0.77147	222.7509	114.7587	12.7
SPEA	0.76702	223.1221	115.1808	12.6
CNSGA2	0.76814	223.3169	114.8438	12.4
NSGA2	0.7675	223.6509	114.8642	12.5

B. Comparison among selection policies

Tables VII to IX show the results obtained by each policy of selection. The columns labeled with *maximum* are the average of the maximums; i.e., the average of the maximums in ten runs is computed for each parameter. Note that static policies outperformed the other policies in the primary objective, but are beaten in the other objectives. It can be also noted that, although the seven policies accepted all traffic request in the low load scenario, no one was able to route all the traffic requests in the other two scenarios. Clearly, static policy with delay objective as primary objective had by far the worst performance considering *Na*. The following notation is used for the last tables.

The best solution for a certain column.
The second better solution for a certain column.
The third better solution for a certain column.

TABLE VII. EXPERIMENTAL RESULTS FOR SCENARIO 1: LOW LOAD.

	Averages				Maximums			
	$\overline{\alpha_p}$	$\overline{C_p}$	$\overline{D_p}$	Na	α	C	D	Na
$\alpha_T D_A$	0.09877	25.7748	121.843	0	0.196	54.55	203.828	0
$C\alpha$	0.11865	20.6330	113.189	0	0.233	41.8	170.098	0
αC	0.09528	23.6273	128.947	0	0.175	50.325	231	0
$D_A\alpha$	0.14108	25.2554	103.48	0	0.267	52.65	146.425	0
SE1	0.14113	25.2465	103.508	0	0.267	52.725	146.306	0
SE2	0.14106	25.2387	103.507	0	0.267	52.75	146.298	0
SE3	0.14114	25.2584	103.509	0	0.267	52.65	146.369	0
DP	0.1347	21.0742	107.876	0	0.25	42.7	153.902	0
DC	0.10278	22.2371	113.971	0	0.2	46.9	165.568	0

TABLE VIII. EXPERIMENTAL RESULTS FOR SCENARIO 2: HIGH LOAD.

	Averages				Maximums			
	$\overline{\alpha_p}$	$\overline{C_p}$	$\overline{D_p}$	Na	α	C	D	Na
$D_A\alpha$	0.66457	167.59	102.71	1.9	1	376.02	145.36	2
$\alpha_T D_A$	0.51312	166.78	115.04	1	0.933	378.47	183.91	1
αC	0.49473	151.11	123.13	1	0.892	337.3	178.19	1
$C\alpha$	0.57736	140.10	115.52	1	0.992	308.87	157.69	1
SE1	0.55270	145.45	116.05	1	0.896	335.9	155.76	1
SE2	0.59061	150.69	112.13	1	0.925	359.17	147.82	1
SE3	0.58980	146.60	112.87	1	0.921	342.15	154.88	1
DP	0.63564	141.70	111.99	1	1	316.2	149.36	1
DC	0.534	147.63	110.12	1	1	329.1	159.87	1

TABLE IX. EXPERIMENTAL RESULTS FOR SCENARIO 3. SATURATION.

	Averages				Maximums			
	$\overline{\alpha_p}$	$\overline{C_p}$	$\overline{D_p}$	Na	α	C	D	Na
$D_A\alpha$	0.8458	257.45	101.10	22.67	1	404.02	122.6	25
$\alpha_T D_A$	0.7111	259.74	110.70	12.72	1	415.55	137.8	14
αC	0.6951	236.80	118.49	12.2	1	385.92	148.2	13.3
$C\alpha$	0.7685	223.21	114.91	12.55	1	356.85	141.6	13.3
SE1	0.7183	232.94	116.88	12.22	1	378.67	143.6	13
SE2	0.7442	235.65	114.54	12.32	1	386.02	138.2	13
SE3	0.7474	231.61	115.20	12.15	1	372	139.0	12.8
DP	0.7889	225.34	113.28	12.5	1	357.6	135.1	13
DC	0.7404	232.91	107.79	13.2	1	377.5	133.9	14

In order to find a policy with the best trade-off value, it is necessary to define a metric that takes into account the three objectives in a combined way. Equation (13) defines *Y*, which is used to compare the different selection policies.

$$Y = \sqrt{\frac{\left(\overline{a_p}/\overline{a_{p\,\max}}\right)^2 + \left(\overline{C_p}/\overline{C_{p\,\max}}\right)^2 + \left(\overline{D_p}/\overline{D_{p\,\max}}\right)^2}{3}}, \qquad (13)$$

where $\overline{a_{p\,\max}}$, $\overline{C_{p\,\max}}$ and $\overline{D_{p\,\max}}$ are the maximum average value of each objective obtained in a given scenario; e.g., $\overline{a_{p\,\max}}$ = 0,1414 for scenario 1, and a point (0.09677, 25.7748, 121.843) is mapped to *Y*= 0.84022. Note that, with this criterion, the nearest solution to the point (0, 0, 0) is considered the best option. Table X shows the values for each solution of Tables VII to IX. While static and semi-static policies choose solutions located in extremes of the Pareto front, *DC* policy, with its selection mechanism, allows a trade-off relation among the three objectives. Hereby, it obtains the best performance when metric *Y* was considered. Moreover, it does not need an a priori parameter.

VI. CONCLUSION AND FUTURE WORKS

This paper presents an empirical comparison among MOEAs to solve the MTE problem for computer networks and a set of selection policies to choose the most suitable solution from the Pareto set in dynamic environments. Simulation results show that the four MOEAs obtain quite similar performance indicating that any of them may be chosen; however, the selection policy is determinant for performance. Among the selection policies, the *DC* criterion finds the solution with the best trade-off relation among the maximum link utilization, average delay and cost of the tree. In summary, the most important key that should be taken into account in MTE is the selection policy, rather than the MOEA to be implemented.

In a future work, the authors will considerer a traffic engineering scheme using multiple trees, where the data flow of a multicast group is transmitted to the destinations through several trees.

TABLE X. METRIC Y FOR DIFFERENTE SCENARIOS.

	Low load	High Load	Saturation	Average
DC	0.82780	0.86048	0.89403	0.86077
$C\alpha$	0.84021	0.88201	0.91371	0.87863
αC	0.87485	0.88831	0.91406	0.89241
DP	0.8716	0.90493	0.91954	0.89870
αD_A	0.89114	0.90545	0.92734	0.90797
SE1	0.931612	0.881902	0.912617	0.908710
SE3	0.93178	0.893160	0.916774	0.913906
SE2	0.931323	0.899511	0.918622	0.916485
$D_A\alpha$	0.931496	0.947951	0.950549	0.94333

REFERENCES

[1] J. Crichigno, and B. Baran, “Multiobjective multicast routing algoritm for traffic enegineering,” *IEEE ICCCN* 2004, Chicago USA.

[2] Y. Donoso, R. Fabregat, F. Solano, J. L. Marzo & B. Baran, “Generalized multiobjective multitree model for dynamic multicast groups,” to be published in *IEEE ICC* 2005, Seul Corea.

[3] A. Roy, N. Banerjee, S. Dias, “An efficient multi-objective QoS-routing algorithm for wireless multicasting,” *INFOCOM* 2002.

[4] X. Cui, C. Lin, and Y. Wei, “A multiobjective model for QoS multicast routing based on genetic algorithm,” *ICCNMC* 2003.

[5] A. Roy, and S. Das., “QM2RP: A QoS-based mobile multicast routing protocol using multi-objective genetic algorithm,” *Journal of Mobile Comm., Computation and Information, Wireless Networks*, May 2004.

[6] J. Crichigno, and B. Baran, “A multicast routing algorithm using multiobjective optimization,” *IEEE ICT’2004*, Ceara, Brasil, 2004.

[7] J. Crichigno and B. Baran, “Multiobjective multicast routing algorithm,” *IEEE ICT’2004*, Ceará, Brasil, 2004.

[8] J. Crichigno, F. Talavera, J. Prieto and B. Barán, “Enrutamiento multicast utilizando optimización multiobjetivo,” *CACIC’2004*, Buenos Aires, Argentina, 2004. pp. 147-158.

[9] F. Talavera, J. Prieto, J. Crichigno and B. Barán. “Comparación de algoritmos evolutivos multi-objetivos en un ambiente multicast,” *CACIC’2004*, Buenos Aires, Argentina, 2004. pp. 1611-1622.

[10] R. Frabegat, Y. Donoso, F. Solano, and J.L. Marzo, “Multitree routing for multicast flows: a genetic algorithm approach,” *CCIA* 2004.

[11] D. Van Veldhuizen, “Multiobjective evolutionary algorithms: classifications, analysis, and new innovations,” *Ph.D. dissertation*, School of Engineering, Air Force Institute of Technology.

[12] C. Coello Coello. Evolutionary multiobjetive optimization: current and future challenges. http://www.lania.mx/~ccoello/EMOO/

[13] 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.

[14] E. Zitzler, Marco Laumanns, and Lothar Thiele, “SPEA2. Improving the strength Pareto evolutionary algorithm for multi-objetive optimization,” http://www.lania.mx/~ccoello/EMOO/

[15] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. “A Fast elitist non-dominated sorting genetic algorithm for multi-objetive optimization: NSGAII,” http://www.lania.mx/~ccoello/EMOO.

[16] K. Deb, and T. Goel. “Controlled elitist non-dominated sorting genetic algorithm for better convergence,” http://www.lania.mx/~ccoello/EMOO/.