

# Enrutamiento Multicast Multiobjetivo basado en Colonia de Hormigas

Diego Pinto & Benjamín Barán

Centro Nacional de Computación, Universidad Nacional de Asunción,  
San Lorenzo, Paraguay  
{dpinto,bbaran}@cnc.una.py

Resumen. Este trabajo propone la resolución del problema de Enrutamiento Multicast en un contexto Multiobjetivo con tres tradicionales algoritmos basados en Colonia de Hormigas. Estos fueron re-estructurados para optimizar simultáneamente cuatro funciones objetivos (la máxima utilización de los enlaces, costo, retardo medio y máximo de fin a fin), obteniendo un conjunto de soluciones óptimas o Frente Pareto sin restricciones a priori, en una sola corrida. Se realizaron simulaciones estáticas como también semi-dinámicas, comparando los resultados con un novedoso algoritmo evolutivo diseñado exclusivamente para el problema en cuestión. Con este trabajo se concluye que las nuevas versiones basadas en Colonia de Hormigas son alternativas válidas para el problema de Enrutamiento Multicast Multiobjetivo al lograr buenos resultados en comparación al algoritmo evolutivo.

## 1 Introducción

Una transmisión *Multicast* consiste en el envío simultáneo o concurrente de datos desde una fuente a un conjunto de nodos destinos componentes de una red de computadoras [1]. En estos últimos años el problema Enrutamiento Multicast, que es tratada por la Ingeniería de Tráfico Multicast, se ha vuelto más importante debido a la creciente utilización de nuevas aplicaciones de transmisión punto a multipunto, como lo son la transmisión de radio y televisión, vídeo bajo demanda, tele conferencias y enseñanza a distancia, las cuáles necesitan requerimientos mínimos que garanticen una adecuada Calidad de Servicios.

A partir de esta creciente tendencia, el retardo desde la fuente a cada uno de los destinos se torna una variable de vital importancia en transmisiones *Multicast* de audio y/o vídeo [2]. Otro punto importante en la Ingeniería de Tráfico Multicast son los "costos" del árbol, entendiéndose por "costos" otras métricas a ser minimizadas como: el número de saltos (*hop count*), utilización del ancho de banda, etc. De esta forma, la Ingeniería de Tráfico Multicast queda planteada como un *problema multiobjetivo*.

Optimización por Colonia de Hormigas (*Ant Colony Optimization* - ACO) es una meta-heurística, propuesta por Dorigo et al. [3], que se inspira directamente en el comportamiento de las colonias reales de hormigas para solucionar problemas de

optimización combinatoria. ACO se basan en una colonia de hormigas artificiales, esto es, unos agentes computacionales simples que trabajan de manera cooperativa y se comunican indirectamente mediante rastros de feromona artificiales.

Los algoritmos de ACO son esencialmente *algoritmos constructivos*: en cada iteración del algoritmo, cada hormiga construye una solución al problema recorriendo un grafo de construcción. Cada arista del grafo, que representa los posibles pasos que una hormiga puede dar, tienen asociados dos tipos de información que guían el movimiento de cada hormiga:

- *La Visibilidad*, que mide la preferencia heurística de moverse desde un nodo a otro nodo teniendo en cuenta valores de costos.
- *Los rastros de feromona artificiales*, que mide la "deseabilidad aprendida" del movimiento de un nodo a otro nodo. Imita a la feromona real que depositan las hormigas. Esta información se modifica durante la ejecución del algoritmo, dependiendo de las soluciones que van siendo encontradas por las hormigas.

En este trabajo se presenta por primera vez, en un contexto multiobjetivo, la reestructuración de tres reconocidos algoritmos basados en Colonia de Hormigas para la resolución del problema de Enrutamiento Multicast, y ellos son:

- *Multiobjective Max-Min Ant System* (M3AS), basado en el conocido *Max-Min Ant System* (MMAS) que demostró su excelente desempeño en problemas mono-objetivos, especialmente en la resolución del problema del cajero viajante (*Traveling Salesman Problem* - TSP) y asignación cuadrática (*Quadratic Assignment Problem* - QAP) [4].
- *Multiobjective Omicron ACO* (MOA), basado en el novedoso *Omicron ACO* (OA) que experimentalmente obtuvo excelente desempeño en problemas mono-objetivos (incluso superando el desempeño al MMAS) especialmente en el TSP [5].
- *Multi-Objective Ant Colony System* (MOACS) [6], con resultados experimentales que demostraron que es el mejor algoritmo multiobjetivo basado en colonia de hormigas para el problema TSP bi-objetivo [7].

A modo de comprobar los resultados obtenidos con los ACOs (M3AS, MOA, MOACS), fueron realizadas pruebas experimentales (simulaciones) con varios grupos *multicast* en un ambiente estático y semi-dinámicos (con varios niveles de carga en la red) sobre la conocida red *Nippon Telegraph and Telephone, Co.* (NTT) de Japón [8]. Los resultados obtenidos fueron comparados con los obtenidos del algoritmo *Multiobjective Multicast Algorithm* (MMA) [9, 10, 11], que está basado en el algoritmo evolutivo de segunda generación *Strength Pareto Evolutionary Algorithm* (SPEA) [12]. Cabe destacar que el MMA fue recientemente publicado y representa uno de los exponentes del estado del arte en la construcción del árbol *multicast* en el contexto multiobjetivo.

En lo que resta este trabajo queda organizado de la siguiente forma: en la Sección 2 se presenta el Problema de Optimización Multiobjetivo, quedando la formulación matemática del Problema de Enrutamiento Multicast en la Sección 3. En la Sección 4 se presentan las nuevas propuestas basadas en ACO. El ambiente experimental es presentado en la Sección 5 quedando expuestos los resultados experimentales en la Sección 6. Finalmente las conclusiones y trabajos abiertos son dados en la Sección 7.

## 2 Problema de Optimización Multiobjetivo

Un Problema de Optimización Multiobjetivo (*Multiobjective Optimization Problem* – MOP) general incluye un conjunto de  $n$  parámetros (variables de decisión), un conjunto de  $k$  funciones objetivo, y un conjunto de  $m$  restricciones. Las funciones objetivos y las restricciones son funciones de las variables de decisión. Luego, el MOP puede expresarse como:

$$\begin{array}{ll} \text{Optimizar} & y = f(x) = (f_1(x), f_2(x), \dots, f_{|y|}(x)) \\ \text{sujeto a} & e(x) = (e_1(x), e_2(x), \dots, e_{|e|}(x)) \geq 0 \\ \text{donde} & x = (x_1, x_2, \dots, x_{|x|}) \in X \\ & y = (y_1, y_2, \dots, y_{|y|}) \in Y \end{array} \quad (1)$$

siendo  $x$  el vector de decisión donde  $|x|$  representa su cardinalidad, mientras que  $y$  es el vector objetivo con cardinalidad representada por  $|y|$ . El espacio de decisión se denota por  $X$ , y el espacio objetivo por  $Y$ . Optimizar, dependiendo del problema, puede significar igualmente, minimizar o maximizar. El conjunto de restricciones  $e(x) \geq 0$ , cuya cardinalidad es representado por  $|e|$ , determina el conjunto de soluciones factibles  $X_f \subset X$  y su correspondiente conjunto de vectores objetivo factibles  $Y_f \subset Y$ .

El problema de optimización con un solo objetivo (*Single Optimization Problem* – SOP) consiste en hallar la  $x$  que tenga el "mejor valor" de  $f(x)$ . Dentro del contexto multiobjetivo no existe un único mejor valor, sino un "conjunto de soluciones óptimas". Entonces, un nuevo concepto de optimalidad debe ser establecido para MOPs. Dado dos vectores de decisión  $u, v \in X_f$  en un contexto de minimización, se puede establecer la siguiente notación:

$$\begin{array}{ll} f(u) = f(v) & \text{si y solo si } \forall i \in \{1, 2, \dots, k\}: f_i(u) = f_i(v) \\ f(u) \leq f(v) & \text{si y solo si } \forall i \in \{1, 2, \dots, k\}: f_i(u) \leq f_i(v) \\ f(u) < f(v) & \text{si y solo si } f(u) \leq f(v) \wedge f(u) \neq f(v) \end{array} \quad (2)$$

En este contexto, esta situación puede darse una de las siguientes tres condiciones, que permiten introducir los símbolos introducidos a continuación:

$$\begin{array}{lll} u > v \text{ (} u \text{ domina a } v\text{)} & \text{sí y solo sí} & f(u) < f(v) \\ v > u \text{ (} v \text{ domina a } u\text{)} & \text{sí y solo sí} & f(v) < f(u) \\ u \sim v \text{ (} u \text{ y } v \text{ no son comparables)} & \text{sí y solo sí} & f(u) \not< f(v) \wedge f(v) \not< f(u) \end{array} \quad (3)$$

Alternativamente,  $u \triangleright v$  denota que  $v > u$  o  $u \sim v$ .

Dado un vector de decisión  $x \in X_f$ , se dice que  $x$  es no dominado respecto a un conjunto  $Q \subseteq X_f$  si y solo si  $x \triangleright v, \forall v \in Q$ . En caso que  $x$  sea no dominado respecto a todo el conjunto  $X_f$  y solo en ese caso, se dice que  $x$  es una solución Pareto óptima. Por lo tanto, el conjunto Pareto óptimo  $X_{true}$  puede ser definido formalmente de la siguiente manera:

$$X_{true} = \{ x \in X_f \mid x \text{ es no dominado con respecto a } X_f \} \quad (4)$$

El correspondiente conjunto de vectores objetivo  $Y_{true} = f(X_{true})$  constituye el Frente Pareto óptimo.

### 3 Formulación del Problema

La red es modelada como un grafo dirigido  $G = (V, E)$ , donde  $V$  es el conjunto de vértices y  $E$  el conjunto de arcos. Los vértices del grafo representan nodos de la red, y los arcos representan los enlaces entre los nodos. Sea:

- $(i, j) \in E$ : Enlace entre los nodos  $i$  y  $j$ ;  $i, j \in V$ .
- $t_{ij} \in \mathfrak{R}^+$ : Tráfico actual del enlace  $(i, j)$ , en Mbps.
- $c_{ij} \in \mathfrak{R}^+$ : Costo del enlace  $(i, j)$ .
- $d_{ij} \in \mathfrak{R}^+$ : Retardo del enlace  $(i, j)$ .
- $z_{ij} \in \mathfrak{R}^+$ : Capacidad del enlace  $(i, j)$ , en Mbps.
- $s \in V$ : Nodo origen de un grupo *multicast*.
- $N_r = \{n_1, n_2, \dots, n_{|N_r|}\}$ : Conjunto de nodos destinos del grupo *multicast*. Donde  $|\cdot|$  indica cardinalidad.
- $\phi \in \mathfrak{R}^+$ : Demanda de tráfico del grupo *multicast*, en Mbps.
- $T(s, N_r)$ : Árbol *multicast* con origen en  $s$  y nodos destinos  $N_r$ .
- $p_T(s, n_i) \subseteq T(s, N_r)$ : Camino que conecta el nodo fuente  $s$  y el nodo destino  $n_i$ .
- $d(p_T(s, n_i))$ : Retardo del camino  $p_T(s, n_i)$ , dado por la suma de los retardos de los enlaces que conforman el camino. Esto es,

$$d(p_T(s, n_i)) = \sum_{(i, j) \in p_T(s, n_i)} d_{ij} \quad (5)$$

Usando las notaciones definidas arriba, el problema de Enrutamiento Multicast puede ser definido como un *MOP* que trata de hallar el árbol *multicast*  $T(s, N_r)$  minimizando las siguientes funciones objetivos:

1. *Máxima utilización del enlace:*

$$\alpha_T = \text{Max}_{(i, j) \in T} \{ (\phi + t_{ij}) / z_{ij} \} \quad (6)$$

2. *Costo del árbol:*

$$C_T = \phi \cdot \sum_{(i, j) \in T} c_{ij} \quad (7)$$

3. *Retardo máximo de extremo a extremo:*

$$DM_T = \text{Max}_{n_i \in N_r} \{ d(p_T(s, n_i)) \} \quad (8)$$

4. *Retardo medio:*

$$DA_T = \frac{1}{|N_r|} \cdot \sum_{n_i \in N_r} d(p_T(s, n_i)) \quad (9)$$

sujeto a restricciones de capacidad en los enlaces:  $\phi + t_{ij} \leq z_{ij} \quad \forall (i, j) \in T(s, N_r)$

Note que  $y \equiv [\alpha_T \ C_T \ DM_T \ DA_T]$  mientras que  $x \equiv T(s, N_r)$ .

## 4 Algoritmos Propuestos basado en ACO

Este trabajo propone resolver el problema de enrutamiento multicast multiobjetivo con algoritmos basados en ACO. Para lo cual fueron seleccionados tres de los tradicionales algoritmos: Max-Min Ant System (MMAS) [4] y Omicron ACO (OA) [5] que son mono-objetivos, como también el Multiobjective Ant Colony System (MOACS) que es multiobjetivo [6]. Dichos algoritmos fueron modificados para el problema en cuestión, donde las características de las versiones originales y de las propuestas son presentadas en la Tabla 1.

Tabla 1. Comparación entre las características de los algoritmos seleccionados y los algoritmos propuestos.

Max-Min Ant System (MMAS)	Multiobjective MMAS (M3AS)
Optimización Mono-objetiva. Búsqueda de la mejor solución global. Niveles máximos y mínimos de feromonas. Elitista – actualización de feromonas con la mejor solución hallada. Heurística de 1 visibilidad.	Optimización Multiobjetiva (4 objetivos). Búsqueda del Frente Pareto. Niveles máximos y mínimos de feromonas. Elitista – actualización de feromonas con el Frente Pareto hallado. Heurística de 3 visibilidades (costo, retardo y tráfico actual).
Omicron ACO (OA)	Multiobjective OA (MOA)
Optimización Mono-objetiva. Búsqueda de la mejor solución global. Mantiene un conjunto ordenado de mejores soluciones. Niveles máximos y mínimos de feromonas. Elitista – actualización de feromonas con el conjunto de ordenados de soluciones. Heurística de 1 visibilidad.	Optimización Multiobjetiva (4 objetivos). Búsqueda del Frente Pareto. Mantiene un conjunto ordenado de Frentes de soluciones. Sin imposición de niveles de feromonas. Elitista – actualización de feromonas con los Frentes de soluciones ordenados. Heurística de 3 visibilidades (costo, retardo y tráfico actual).
Multiobjective Ant Colony System (MOACS)	Multiobjective Ant Colony System (MOACS) <sup>1</sup>
Optimización Multiobjetiva (3 objetivos). Búsqueda del Frente Pareto. Selección pseudo-aleatoria y evaporación de feromonas en línea de los enlaces durante la construcción de una solución. Heurística de 2 visibilidades.	Optimización Multiobjetiva (4 objetivos). Búsqueda del Frente Pareto. Selección pseudo-aleatoria y evaporación de feromonas en línea de los enlaces durante la construcción de una solución. Heurística de 3 visibilidades (costo, retardo y tráfico actual).

Los tres algoritmos propuestos tienen características comunes desde el punto de vista del funcionamiento como optimizadores. A continuación son citadas las similitudes:

1. Búsqueda del Frente Pareto utilizando solo una colonia de hormigas.
2. Búsqueda de soluciones es guiada por las feromonas como también por tres visibilidades (costo, tráfico y retardo) de cada enlace.
3. La probabilidad o deseabilidad de selección de un enlace aún no visitado y que cumpla con las restricciones impuestas por el problema, está dada por el siguiente ecuación:

<sup>1</sup> El MOACS mantiene el mismo nombre por ser un algoritmo multiobjetivo. Sin embargo, note que por primera vez fue utilizado para optimizar cuatro objetivos simultáneamente utilizando 3 visibilidades.

$$P_{ij} = \frac{\tau_{ij} \cdot (\eta_{ij}^c)^{\lambda_c} \cdot (\eta_{ij}^d)^{\lambda_d} \cdot (\eta_{ij}^t)^{\lambda_t}}{\sum_{\forall g \in N_i} \tau_{ij} \cdot (\eta_{ig}^c)^{\lambda_c} \cdot (\eta_{ig}^d)^{\lambda_d} \cdot (\eta_{ig}^t)^{\lambda_t}} \quad (i, j) \in E \quad (10)$$

Donde:

- $\eta_{ij}^c$  visibilidad por el costo del enlace  $(i, j)$ ,
- $\eta_{ij}^d$  visibilidad por el retardo del enlace  $(i, j)$ ,
- $\eta_{ij}^t$  visibilidad por el tráfico actual del enlace  $(i, j)$ ,
- $\beta$  influencia relativa entre las visibilidades y el nivel de feromonas
- $\tau_{ij}$
- $\lambda_c$  influencia relativa de la visibilidad por costo,
- $\lambda_d$  influencia relativa de la visibilidad por retardo,
- $\lambda_t$  influencia relativa de la visibilidad por tráfico actual,
- $N_i$  conjunto de nodos vecinos factibles al nodo  $i$  en la que se encuentra a la hormiga.

4. Actualización de los niveles de feromonas es realizada utilizando soluciones del Frente Pareto. El MOA en particular además utiliza sus otros Frentes de soluciones. Esta actualización esta dada por la siguiente ecuación:

$$\tau_{ij} = \tau_{ij} + \Delta\tau_k \quad \forall (i, j) \in T_k \quad (11)$$

$$\Delta\tau_k = \frac{1}{f(\alpha_{T_k}, C_{T_k}, DM_{T_k}, DA_{T_k})} \quad (12)$$

Donde:

- $T_k$  es la k-esima solución del Frente Pareto
- $\Delta\tau_k$  es el nivel de feromonas agregado al enlace  $(i, j) \in T_k$

Podemos citar que en particular el MOACS se diferencia por utilizar una regla pseudo-aleatoria para la selección de los enlaces, durante la construcción de una solución. Además, la actualización de las feromonas depende del estado del Frente Pareto, es decir, si la colonia no halla nuevas soluciones óptimas las feromonas se actualizan en función de las soluciones actuales del Frente Pareto, en otro caso se re-inicia los niveles de las feromonas [6].

En cambio el M3AS se caracteriza por imponer niveles máximos y mínimos a las feromonas de forma tal a prevenir la convergencia prematura [4].

El MOA se caracteriza por tener varios Frentes de soluciones que se encuentran ordenados por dominancia Pareto. La actualización de las feromonas es realizada en distintas proporciones para cada Frente. Es decir, las soluciones del primer Frente aportan más feromonas que la segunda, la segunda más que la tercera, etc.

La Figura 1 presenta el esquema general del algoritmo M3AS. En la Figura 2 es mostrado el esquema correspondiente al MOA quedando finalmente expuesto el MOACS en la Figura 3.

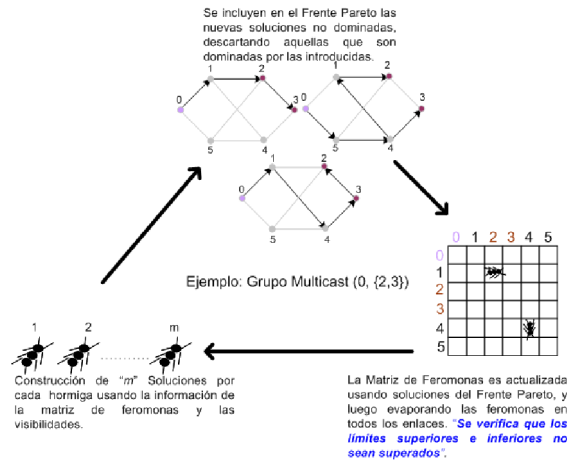


Fig. 1. Esquema general del M3AS. Este algoritmo utiliza  $m$  hormigas para construir  $m$  soluciones utilizando la información de la matriz de feromonas y las visibilidades. Cada una de las soluciones es evaluada con respecto al Frente Pareto, si la solución es no dominada ésta es incluida siendo descartadas aquellas dominadas. Luego con las soluciones actuales del Frente Pareto es actualizada la matriz de feromonas limitado por un nivel máximo. Después de esta actualización se procede a evaporar las feromonas verificando que los niveles mínimos no sean superados. Este proceso sigue hasta que algún criterio de parada sea satisfecho.

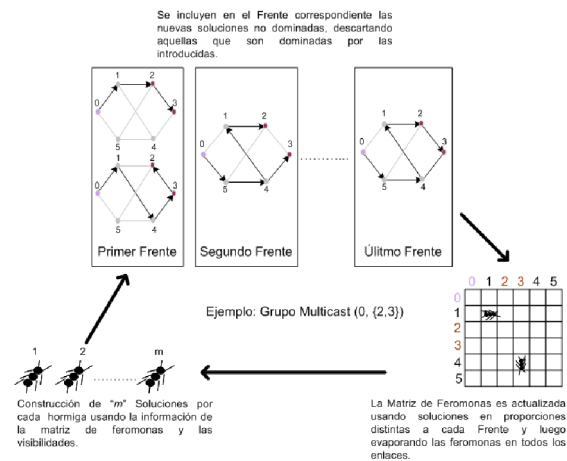


Fig. 2. Este algoritmo MOA utiliza  $m$  hormigas para construir  $m$  soluciones utilizando la información de las feromonas y las visibilidades. Cada una de las soluciones es incluida en el Frente en que no es dominada, eliminando aquellas soluciones que dominada. Luego con las soluciones actuales de los Frentes es actualizada la matriz de feromonas con una proporción distinta para cada Frente. Después de esta actualización se procede a evaporar las feromonas en todos los enlaces. Este proceso sigue hasta que algún criterio de parada sea satisfecho.

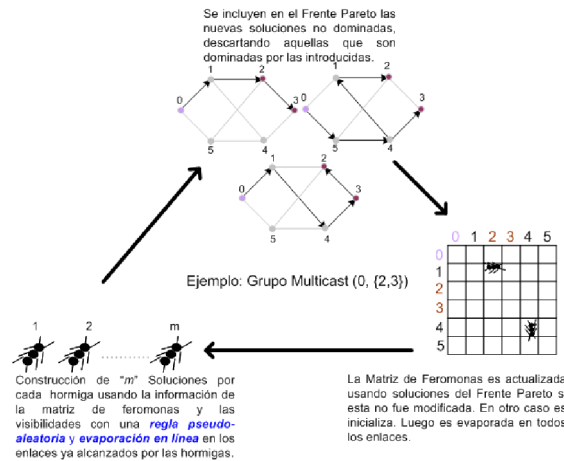


Fig. 3. El esquema general del MOACS se caracteriza por utilizar una colonia de  $m$  hormigas para construir  $m$  soluciones. Al término del recorrido de cada hormiga se evaporan los enlaces recorridos pertenecientes a la nueva solución. La actualización del Frente Pareto es igual que los anteriores algoritmos. La actualización de las feromonas es llevada a cabo si el Frente Pareto no fue modificado, en otro caso las feromonas son re-inicializadas. Este proceso sigue hasta que algún criterio de parada sea satisfecho.

Note que los algoritmos anteriores son independientes al tipo de problema combinatorio a resolver. Para poder trasladarlo al problema que se desee, debe ser planteado un algoritmo constructor de una solución que represente el funcionamiento de cada hormiga de la colonia. En el caso particular del problema de enrutamiento multicast cada hormiga debe construir un árbol de expansión para el grupo multicast  $(s, N_r)$ . Dicho algoritmo que representa a la "hormiga multicast" fue desarrollado en el marco de este trabajo y es presentado en la Figura 4.

```

Inicio Construir Árbol
Leer  $\lambda^c, \lambda^d, \lambda^s, \alpha, \beta, \phi, \tau_{ij}, (s, N_r)$ 
 $T = \emptyset$       /* Árbol en construcción
 $N = "s"$      /*  $N$  conjunto de nodos de partida
Repetir hasta que todos los destinos de  $N_r$  sean alcanzados{
  Seleccionar aleatoriamente un nodo  $i$  de  $N$ 
  Crear lista de nodos vecinos factibles  $N_i$  al nodo  $i$ 
  Calcular las probabilidades  $p_{ij}$  de la lista  $N_i$  según ecuación (10)
  Seleccionar nodo  $j$  utilizando Regla pseudo-aleatorio para MOACS o
  seleccionar aleatoriamente nodo  $j$  para M3AS y MOA.
   $T = T \cup (i, j)$  /* Enlace  $(i, j)$  adherido al árbol  $T$ 
   $N = N \cup j$  /* Nodo  $j$  adherido al conjunto de partida
} Fin repetir
Podar Árbol  $T$  /* Se borran todos los enlaces no utilizados */
Retornar  $T$ 
Fin Construir Árbol
  
```

Fig. 4. Algoritmo Construir Árbol. Inicialmente la "hormiga multicast" se ubica en el nodo fuente "s". Luego en el ciclo principal construye una lista de nodos vecinos factibles no visitados asignando probabilidades de selección. Dependiendo del optimizador la hormiga selecciona el enlace con una regla pseudo-aleatoria para el MOACS y aleatoriamente para el M3AS y MOA teniendo en cuenta las probabilidades asignadas. Este proceso sigue hasta alcanzar todos los destinos de  $N_r$  para luego eliminar los enlaces no utilizados y retornar una solución válida.



## 5 Ambiente Experimental

Se realizaron pruebas experimentales sobre la conocida topología de red NTT de Japón [8], con varios tiempos de corrida y varios grupos *multicast*, los cuales son presentados en la Tabla 2. Estas pruebas se dividen en Estáticas y Semi-Dinámicas.

En las *pruebas estáticas* se supone la red se encuentra totalmente descargada (sin tráfico), y los algoritmos deben responder a un solo pedido de grupo *multicast*. Esto se hace para verificar el correcto funcionamiento de los algoritmos y conocer su comportamiento con muy baja carga. El caso *semi-dinámicos* consistirá de pruebas con tres niveles de carga diferentes en la red (baja, media y alta) respectivamente. Los algoritmos deberán responder a un solo pedido de grupo *multicast* por vez. Este escenario es típico de las redes ópticas, donde los cambios de los grupos *multicast* no ocurren muy frecuentemente [13].

Los resultados de las pruebas experimentales también fueron comparados con el MMA [9, 10, 11], el cuál es un algoritmo evolutivo, diseñado especialmente para el problema Enrutamiento Multicast Multiobjetivo.

Tabla 2. Grupos Multicast utilizados en todas las pruebas.

GM*	$(s, N_r)$	$ N_r $
1	{5}{0, 1, 8, 10, 22, 32, 38, 43, 53}	9
2	{4}{0, 1, 3, 5, 9, 10, 12, 23, 25, 34, 37, 41, 46, 52}	14
3	{4}{0, 1, 3, 5, 6, 9, 10, 12, 17, 22, 23, 25, 34, 37, 41, 46, 47, 52, 54}	19
4	{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

\*GM = Grupo Multicast.

Debido que el frente óptimo Pareto real  $Y_{true}$  necesita un tiempo computacional muy elevado, con lo que la búsqueda se vuelve impráctica, es necesario un frente Pareto aproximado  $Y_{apr}$  con la cuál realizar las comparaciones de cada resultado. Los pasos realizados para obtener  $Y_{apr}$  para cada grupo *multicast* en cada prueba son:

1. Cada algoritmo (M3AS, MOA, MOACS & MMA) fue ejecutado cinco veces y un promedio fue calculado para comparación a cada uno de los otros.
2. Para cada algoritmo, cinco conjunto de soluciones no-dominadas:  $Y_1, Y_2, \dots, Y_5$  fueron calculados, uno para cada corrida.
3. Para cada algoritmo una súper población  $Y_{SP}$  fue obtenida, donde
$$Y_{SP} = \bigcap_{i=1}^5 Y_i$$
4. Soluciones dominadas fueron borradas de  $Y_{SP}$ , obteniendo el frente Pareto calculado por cada algoritmo, como sigue:
  - $Y_{M3AS}$  (Frente Pareto obtenido con las cinco corridas, usando M3AS)
  - $Y_{MOA}$  (Frente Pareto obtenido con las cinco corridas, usando MOA)
  - $Y_{MOACS}$  (Frente Pareto obtenido con las cinco corridas, usando MOACS)
  - $Y_{MMA}$  (Frente Pareto obtenido con las cinco corridas, usando MMA)
5. Un conjunto de soluciones  $Y^*$  fue obtenido como:
$$Y^* = Y_{M3AS} \cup Y_{MOA} \cup Y_{MOACS} \cup Y_{MMA}$$
6. Las soluciones dominadas fueron eliminadas de  $Y^*$ , y una aproximación de  $Y_{true}$ , llamado  $Y_{apr}$ , fue finalmente creado.

La Tabla 3 presenta la cantidad de soluciones de  $Y_{apr}$  de cada grupo *multicast* (GM) para cada nivel de carga de la red.

Tabla 3. Número de soluciones pertenecientes a  $Y_{apr}$  de cada grupo multicast.

	GM 1	GM 2	GM 3	GM 4	Pruebas	
	9	9	24	18	Estáticas	
$ Y_{apr} $	15	15	30	34	Baja Carga	Semi Dinámicas
	31	31	63	56	Media Carga	
	46	46	143	115	Alta Carga	

Los métodos de comparación en cada experimento fueron comparados con la cantidad de soluciones contribuidas a  $Y_{apr}$  y la cobertura de soluciones entre algoritmos. La cobertura indica que cantidad de soluciones dominadas entre algoritmos. Las Tablas de comparaciones y coberturas usan las siguientes notaciones:

$|Y_{alg}|$  es el promedio del número de soluciones del frente Pareto calculado por cada algoritmo. Ejemplo para el M3AS,  $|Y_{alg}| = |Y_{M3AS}|$ .

$|Y_{apr} \cap Y_{alg}|$  es el promedio del número de soluciones que pertenecen a la vez al frente Pareto calculado por cada algoritmo  $Y_{alg}$  y también en el frente Pareto aproximado  $Y_{apr}$ .

$|Y_{apr} \succ Y_{alg}|$  es el promedio del número de soluciones del frente Pareto calculado por cada algoritmo  $Y_{alg}$  que son dominados por el frente Pareto aproximado  $Y_{apr}$ .

$\%Y_{apr}$  es el promedio del porcentaje de soluciones halladas del frente Pareto aproximado  $Y_{apr}$ , i.e.  $100 \cdot (|Y_{apr} \cap Y_{alg}|) / (|Y_{apr}|)$ .

## 6 Resultados Experimentales

En esta sección son presentados los resultados globales para cada grupo multicast en cada prueba (estática y semi-dinámica). Dichos resultados están presentados en las Tablas 4 y 5 respectivamente.

Tabla 4. Promedio Global de las Pruebas Estáticas.

	Comparación de Soluciones con $Y_{apr}$				Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\%Y_{apr}$	$Y_{M3AS}$	$Y_{MOA}$	$Y_{MOACS}$	$Y_{MMA}$
$Y_{M3AS}$	9.21	3.14	12.35	58.44		2.35	0	0.16
$Y_{MOA}$	7.63	3.85	11.48	49.69	1.08		0.05	0.10
$Y_{MOACS}$	13.71	2.40	16.11	82.34	2.08	3.18		0.29
$Y_{MMA}$	8.20	1.16	9.36	49.27	1.25	1.89	0.14	

Se puede concluir brevemente en estas pruebas que el MOACS fue el mejor por tener mayor porcentaje de soluciones halladas del Frente Pareto (82.34%) y menor cantidad de sus soluciones dominadas con respecto al resto de los algoritmos (0, 0.05 y 0.14). Además, el MOA y el M3AS tienen mayor porcentaje de soluciones

pertenecientes al  $Y_{apr}$  (49.69% y 58.44%) en comparación al MMA que solo alcanzó un 49.27%.

Tabla 5. Promedio Global de las Pruebas Dinámicas.

	Comparación de Soluciones con $Y_{apr}$				Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\%Y_{apr}$	$Y_{M3AS}$	$Y_{MOA}$	$Y_{MOACS}$	$Y_{MMA}$
$Y_{M3AS}$	10.89	9.75	20.64	21.66		4.57	0	1.67
$Y_{MOA}$	9.45	9.32	18.77	18.68	3.08		0.36	1.78
$Y_{MOACS}$	32.18	7.76	39.94	56.59	6.70	7.98		2.44
$Y_{MMA}$	8.11	3.91	12.02	16.82	4.51	5.09	4.11	

En estas pruebas puede notarse que el MOACS obtuvo el mayor porcentaje de soluciones pertenecientes al  $Y_{apr}$  (56.59%), además obtuvo mayor cobertura sobre M3AS y MOA (6.7 y 7.98 respectivamente). Sin embargo, puede notarse que el MMA logró dominar más soluciones del MOACS (4.11), mientras que sus soluciones fueron dominadas en solo 2.44. El segundo mejor algoritmo fue el M3AS con el 21.66% de las soluciones alcanzadas, quedando en tercer lugar el MOA con 18.68% incluso superior al logrado por el MMA con solo 16.82%. Claramente con estas soluciones notamos que hasta el algoritmo basado en ACO con más pobre desempeño es mejor que el MMA.

## 7 Conclusiones, Aportes y Trabajos Futuros

En años recientes ha surgido una novedosa metaheurística conocida como *Ant Colony Optimization* – ACO que fue exitosamente utilizado en varios problemas combinatoriales, y por primera vez este trabajo propone utilizarlo en la resolución del problema de *enrutamiento multicast multiobjetivo*, donde se optimizan simultáneamente cuatro funciones objetivos: utilización máxima de los enlaces  $\alpha_T$ , costo del árbol  $C_T$ , retardo medio  $DA_T$  y máximo  $DM_T$ . Desde el punto de vista teórico, esto constituye un claro aporte a las Ciencias de Computación, pues se propone una nueva alternativa de solución basado en ACO, debido que el problema solo ha sido resuelto en un contexto puramente multiobjetivo con los algoritmos evolutivos [1, 2, 3] [9, 10, 11].

Para resolver este problema se propuso re-estructurar tres de los mejores algoritmos basado en ACO: "*Max-Min Ant System* - MMAS", "*Ómicrom ACO* - OA" y el "*Multiobjective Ant Colony Optimization* - MOACS". Debido que el MMAS y el OA son algoritmos mono-objetivos, este trabajo propone por primera vez una versión multiobjetivo, por lo cual estos fueron re-nombrados como "*Multobjective Max-Min Ant System* – M3AS" y "*Multiobjective Ómicrom ACO* - MOA" respectivamente. Además el MOACS es utilizado por primera vez en la optimización de cuatro objetivos simultáneamente, para lo cual fueron necesarios otros ajustes importantes en su estructura funcional. Desde el punto de vista de la metaheurística ACO esto constituye un aporte significativo al proponer por primera vez la versión multiobjetivo de varios algoritmos mono-objetivos.

Los algoritmos propuestos fueron evaluados sobre varios grupos *multicast* de distintos tamaños en la red NTT de Japón. Además las pruebas se dividieron en Pruebas Estáticas y Semi-Dinámicas, en las cuales los algoritmos deben hallar un conjunto entero de soluciones óptimas Pareto para cada grupo *multicast*. En las Pruebas Estáticas la red se encuentra vacía y el objetivo es verificar el funcionamiento de los algoritmos a muy baja carga en la red. En las pruebas Semi-Dinámicas la red se encuentra con tráfico inicial. En estas pruebas se tomaron tres niveles de tráfico inicial: baja carga (0 - 0.4 de utilización de enlaces), media carga (0.4 - 0.7 de utilización de enlaces) y saturación (0.7 - 1 de utilización de enlaces).

Para validar las nuevas alternativas, éstas fueron comparadas con el "*Multiobjective Multicast Algorithm - MMA*", el cual es un algoritmo genético basado en un algoritmo evolutivo de segunda generación "*Strength Pareto Evolutionary Algorithm - SPEA*". El MMA fue diseñado exclusivamente para el problema de enrutamiento *multicast* en un contexto puramente multiobjetivo.

Se puede concluir según los resultados, tanto de las extensas Pruebas Estáticas como de las Semi-Dinámicas, que las nuevas versiones M3AS, MOA & MOACS son buenas alternativas para la resolución del problema de Ingeniería de Tráfico Multicast al obtener en general buenas soluciones en comparación al MMA. Quedando claro el mejor desempeño fue logrado por el MOACS, por lo cual este trabajo sugiere como la mejor alternativa a ser utilizada en el enrutamiento *multicast*.

De forma a continuar con el trabajo iniciado, los siguientes tópicos son propuestos como trabajos futuros:

- Evaluación comparativa de M3AS, MOA, MOACS con otros algoritmos multi-objetivos.
- Utilización de las propuestas ACOs a problema de enrutamiento *multicast* sobre redes ópticas.
- Considerando que la optimización con más de cuatro funciones objetivos no es trivial en los enfoques ACO, sería muy interesante optimizar mayor cantidad de funciones objetivos como: jitter, tamaño de paquetes encolados, números de saltos, etc.
- Estudio e implementación de las nuevas propuestas en esquemas con múltiples árboles y múltiples flujos.

## Referencias

1. A. Tanenbaum, Computer Networks, Prentice Hall 4<sup>o</sup> Edition, 2003.
2. P. Ravikumar, y R. Bajpai, "Source-based delay bounded multicasting in multimedia networks", Computer Communications, Vol. 21, 1998, páginas 126-132.
3. M. Dorigo y G. Di Caro. "The Ant Colony Optimization meta-heuristic". En D. Corne, M. Dorigo, y F. Glover, editores, New Ideas in Optimization, páginas 11-32. McGraw Hill, London, UK, 1999.
4. T. Stützle y H. Hoos. "MAX-MIN Ant System". Future Generation Computer System. 16(8): páginas 889-914, junio 2000.
5. O. Gómez y B. Barán. "Ómicron ACO". En Proceedings de la Conferencia Latinoamericana en Informática (CLEI'04). Septiembre 2004.

6. M. Schaerer y B. Barán. "A Multiobjective Ant Colony System For Vehicle Routing Problem With Time Windows", IASTED International Conference on Applied Informatics, Innsbruck, Austria, 2003.
7. C. García-Martínez, O. Cordón y F. Herrera, "An Empirical Analysis of Multiple Objective Ant Colony Optimization Algorithms for the Bi-criteria TSP". En M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, y T. Stützle, editores, Proceedings of ANTS 2004 -Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence. Tomo 3172 de LNCS. Springer-Verlag, Bruselas, septiembre 2004.
8. R. Sosa y B. Barán, "A New Approach for Antnet Routing", IEEE International Conference on Computer Communication and Networks (ICCCN'2000), USA, 2000.
9. J. Crichigno y B. Barán. "Multiobjective Multicast Routing Algorithm for Traffic Engineering". IEEE ICCCN'2004, Chicago, US, 2004.
10. J. Crichigno y B. Barán. "Multiobjective Multicast Routing Algorithm". IEEE ICT'2004, Ceará, Brasil, 2004.
11. J. Crichigno y B. Barán, "A Multicast Routing Algorithm Using Multiobjective Optimization". IEEE ICT'2004, Ceará, Brasil, 2004.
12. E. Zitzler, y L. Thiele, "Multiobjective Evolutionary Algorithms: A comparative Case Study and the Strength Pareto Approach", IEEE Trans. Evolutionary Computation, Vol. 3, No. 4, 1999, páginas 257-271.
13. Y. Yang, J. Wang, C. Qiao. "Nonblocking WDM Multicast Switching Networks". IEEE Computer Society December 2000 (Vol. 11, No. 12) páginas 1274-1287.