

Enrutamiento Multicast Utilizando Optimización Multiobjetivo

Jorge Crichigno^{+,*}, Francisco Talavera⁺, Joel Prieto, Benjamín Barán^{+,*}
jcrichigno@cnc.una.py, ftalavera@yahoo.com.ar, jprieto@telesurf.com.py, bbaran@cnc.una.py

Ciencias y Tecnología - Universidad Católica Nuestra Señora de la Asunción⁺
Centro Nacional de Computación - Universidad Nacional de Asunción^{*}

Resumen

Este trabajo presenta un nuevo algoritmo multi-objetivo, inspirado en un trabajo mono-objetivo anterior, proponiendo la utilización de un algoritmo basado en el SPEA. El algoritmo propuesto optimiza de manera simultánea el costo del árbol, el retardo promedio y el retardo máximo de extremo a extremo. De este modo, un conjunto de soluciones Pareto óptimas es calculado en una sola ejecución del algoritmo sin considerar decisiones a priori.

Resultados experimentales fueron comparados con los conseguidos por otro algoritmo multi-objetivo propuesto anteriormente por algunos autores de este trabajo, mostrándose que para problemas pequeños se llega a las soluciones óptimas en un tiempo menor y para problemas más grandes se puede obtener una mayor cantidad de soluciones con un menor tiempo de procesamiento. Además se presentan simulaciones para el problema dinámico de enrutamiento multicast, donde las solicitudes de tráfico arriban una después de otra. Se compara el rendimiento con el algoritmo SK, el cual encara el problema de enrutamiento multicast mono-objetivo.

Palabras clave: Redes, Algoritmo Evolutivo, Multicast, Multi-objetivo, Pareto.

1 Introducción

Multicast consiste en la transmisión simultánea de datos desde un nodo fuente a un subconjunto de nodos destinos en una red de computadoras [1]. Recientemente, se ha incrementado el interés en los algoritmos de enrutamiento multicast debido a las nuevas aplicaciones punto a multipunto en redes de datos, como transmisiones de radio y TV, video bajo demanda, teleconferencias y aprendizaje a distancia. Dichas aplicaciones generalmente requieren de algunos parámetros de calidad de servicio tales como retardo máximo de extremo a extremo y recursos mínimos de ancho de banda.

Cuando el problema de enrutamiento dinámico considera varias solicitudes de tráfico de distintos nodos, además de los parámetros de calidad de servicio, también se debe tener en cuenta el balanceo de carga y el uso adecuado de los recursos de la red. El enfoque tradicional utilizado para el balanceo de la carga es la minimización de la utilización del enlace más sobrecargado, o minimización de la utilización máxima de los enlaces de la red (α) [2].

Este trabajo propone un algoritmo de enrutamiento multicast multiobjetivo, basado en un Algoritmo Evolutivo de Optimización Multiobjetivo (MOEA) con una población externa de soluciones Pareto óptimas, llamado *Strength Pareto Evolutionary Algorithm* (SPEA) [3]. Dicho algoritmo halla un árbol multicast optimizando varias funciones objetivo simultáneamente.

El presente artículo está organizado de la siguiente manera: en la Sección 2 se describen brevemente los trabajos relacionados. En la Sección 3 se da la definición general de un problema de optimización multiobjetivo. La formulación del problema y las funciones objetivo son dadas en la Sección 4. El algoritmo propuesto es explicado en la Sección 5. Los resultados experimentales son mostrados en la Sección 6. Por último, las conclusiones y los trabajos futuros son presentados en la Sección 7.

2 Trabajos Relacionados

Kompella et al. [5] publican un trabajo interesante sobre el problema de enrutamiento multicast sujeto a retardo máximo de extremo a extremo proponiendo un algoritmo basado en

programación dinámica (algoritmo KPP). Este algoritmo optimiza el costo del árbol multicast y restringe el retardo máximo a un valor dado a priori.

Para el mismo problema planteado por Kompella et al., recientemente han sido propuestos algoritmos evolutivos, una técnica global de búsqueda que emula la evolución natural [6]. Ravikumar et al. [1] propusieron un método basado en un algoritmo genético simple, en el cual la función de adaptabilidad está dada como la inversa del costo del árbol multicast y penalizada cuando el retardo máximo supera la cota superior dada a priori. Este trabajo fue mejorado sucesivamente por Zhengying et al. [7] y Barbosa et al. [8]. La principal desventaja del enfoque basado en restricciones es la necesidad de una cota de retardo máximo definido a priori, que puede descartar soluciones de bajo costo o baja utilización de los enlaces (o ambos) con un retardo máximo apenas superior a la cota predefinida.

La *Ingeniería de Tráfico* permite optimizar los recursos de la red y evitar los cuellos de botella, mediante el encaminamiento explícito y la reserva de recursos a través de los caminos escogidos. En este contexto, Seok et al. propusieron un algoritmo heurístico (denominado SK) basado en una extensión del algoritmo de Dijkstra [2]. El objetivo adoptado en este trabajo es la minimización de la utilización máxima de los enlaces del árbol, considerando como una restricción la longitud de las rutas para evitar caminos muy largos. De esta manera, los autores del algoritmo intentan optimizar, además de la utilización máxima de los enlaces, el consumo de ancho de banda.

En [4], Donoso et al. propusieron un esquema de ingeniería de tráfico multicast multi-árbol. De esta forma, el flujo de datos desde el nodo fuente a los nodos destinos es enviado a través de distintos árboles, de forma a balancear la carga sobre la red.

En [14] y [15], algunos de los autores de este trabajo presentaron un algoritmo de enrutamiento multicast multiobjetivo, que para fines de diferenciación llamaremos MMA1 (*Multiobjective Multicast Algorithm*). Al algoritmo presentado en este trabajo denominaremos MMA2. El MMA1 está basado también en el SPEA, pero se diferencia del nuevo algoritmo aquí propuesto en la forma de codificar y realizar el cruzamiento de los individuos.

3 Problemas de Optimización Multiobjetivo

Un Problema de Optimización Multiobjetivo (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 objetivo y las restricciones son funciones de las variables de decisión. Luego, el MOP puede expresarse como:

$$\begin{array}{ll}
 \text{Optimizar} & \mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})) \\
 \text{sujeto a} & \mathbf{e}(\mathbf{x}) = (e_1(\mathbf{x}), e_2(\mathbf{x}), \dots, e_m(\mathbf{x})) \geq \mathbf{0} \\
 \text{donde} & \mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbf{X} \\
 & \mathbf{y} = (y_1, y_2, \dots, y_k) \in \mathbf{Y}
 \end{array} \tag{3.1}$$

siendo \mathbf{x} el vector de decisión e \mathbf{y} el vector objetivo. El espacio de decisión se denota por \mathbf{X} , y el espacio objetivo por \mathbf{Y} . Optimizar, dependiendo del problema, puede significar igualmente, minimizar o maximizar. El conjunto de restricciones $\mathbf{e}(\mathbf{x}) \geq \mathbf{0}$ determina el conjunto de soluciones factibles $\mathbf{X}_f \subset \mathbf{X}$ y su correspondiente conjunto de vectores objetivo factibles $\mathbf{Y}_f \subset \mathbf{Y}$.

El problema de optimización consiste en hallar la \mathbf{x} que tenga el “mejor valor” de $\mathbf{f}(\mathbf{x})$. En general, no existe un único “mejor valor”, sino un conjunto de soluciones óptimas. Entonces, un nuevo concepto de optimalidad debe ser establecido para MOPs. Dados dos vectores de decisión $\mathbf{u}, \mathbf{v} \in \mathbf{X}_f$, puede darse una de las tres condiciones siguientes:

$$\mathbf{f}(\mathbf{u}) = \mathbf{f}(\mathbf{v}) \text{ si y solo si } \forall i \in \{1, 2, \dots, k\}: f_i(\mathbf{u}) = f_i(\mathbf{v})$$

$$\begin{aligned} \mathbf{f}(\mathbf{u}) \leq \mathbf{f}(\mathbf{v}) \text{ si y solo si } \forall i \in \{1, 2, \dots, k\}: f_i(\mathbf{u}) \leq f_i(\mathbf{v}) & \quad (3.2) \\ \mathbf{f}(\mathbf{u}) < \mathbf{f}(\mathbf{v}) \text{ si y solo si } \mathbf{f}(\mathbf{u}) \leq \mathbf{f}(\mathbf{v}) \wedge \mathbf{f}(\mathbf{u}) \neq \mathbf{f}(\mathbf{v}) & \end{aligned}$$

En un contexto de minimización, esta situación se expresa con los siguientes símbolos y términos:

$$\begin{aligned} \mathbf{u} \succ \mathbf{v} \text{ (} \mathbf{u} \text{ domina a } \mathbf{v} \text{)} & \quad \text{si y solo si} & \quad \mathbf{f}(\mathbf{u}) < \mathbf{f}(\mathbf{v}) \\ \mathbf{v} \succ \mathbf{u} \text{ (} \mathbf{v} \text{ domina a } \mathbf{u} \text{)} & \quad \text{si y solo si} & \quad \mathbf{f}(\mathbf{v}) < \mathbf{f}(\mathbf{u}) \\ \mathbf{u} \sim \mathbf{v} \text{ (} \mathbf{u} \text{ y } \mathbf{v} \text{ no son comparables)} & \quad \text{si y solo si} & \quad \mathbf{f}(\mathbf{u}) \not< \mathbf{f}(\mathbf{v}) \wedge \mathbf{f}(\mathbf{v}) \not< \mathbf{f}(\mathbf{u}) \end{aligned} \quad (3.3)$$

Alternativamente, $\mathbf{u} \triangleright \mathbf{v}$ denota que \mathbf{u} domina o es igual a \mathbf{v} .

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

$$X_{true} = \{ \mathbf{x} \in X_f \mid \mathbf{x} \text{ es no dominado con respecto a } X_f \} \quad (3.4)$$

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

4 Formulación del Problema

La red es modelada como un grafo dirigido $G=(V, E)$, donde V es el conjunto de nodos y E es el conjunto de enlaces. La cardinalidad de cada conjunto será denotada por $|\cdot|$. Sea $(i, j) \in E$ el enlace entre los nodos i y j . Sea z_{ij} , c_{ij} , d_{ij} y $t_{ij} \in \mathbb{R}^+$ la capacidad, el costo, el retardo y el tráfico actual (en bps) del enlace (i, j) . Un grupo multicast está dado por un nodo fuente $s \in V$, un conjunto de nodos destinos $N \subseteq V - \{s\}$ y una demanda de tráfico $\phi \in \mathbb{R}^+$ (en bps). Sea $T(s, N)$ un árbol multicast con origen en s y conjunto de nodos destinos $(N \subset V)$; $p_T(s, n) \subseteq T(s, N)$ representa el camino que conecta el nodo fuente s y el nodo destino $n \in N$, y $d(p_T(s, n))$ el retardo de este camino, dado por la suma de los retardos de los enlaces que conforman el camino. Esto es, $d(p_T(s, n)) = \sum_{(i, j) \in p_T(s, n)} d_{ij}$.

El problema de enrutamiento multicast puede ser formulado como un MOP en el cual se desea construir el árbol multicast $T(s, N)$ que minimice las siguientes funciones objetivo:

$$1\text{- Utilización máxima de los enlaces del árbol: } \alpha_T = \text{Max}_{(i, j) \in T} \{ (\phi + t_{ij}) / z_{ij} \} \quad (4.1)$$

$$2\text{- Costo del árbol: } C_T = \phi \sum_{(i, j) \in T} c_{ij} \quad (4.2)$$

$$3\text{- Retardo máximo de extremo a extremo: } D_M = \text{Max}_{n \in N} \{ d(p_T(s, n)) \} \quad (4.3)$$

$$4\text{- Retardo medio: } D_A = \frac{1}{|N|} \sum_{n \in N} d(p_T(s, n)) \quad (4.4)$$

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

5 Algoritmos Multiobjetivos

El algoritmo propuesto está inspirado en los Algoritmos Genéticos (GA). El algoritmo se inicia con un conjunto de configuraciones aleatorias llamada población inicial. Cada individuo (cromosoma) en la población representa una solución al problema de optimización. En cada

generación, los individuos son evaluados usando una función de adaptabilidad (*fitness*). Basados en este valor, algunos individuos, llamados padres, son seleccionados. La probabilidad de selección de un individuo está relacionada con su adaptabilidad, de forma a asignar mayor probabilidad de selección a los mejores individuos. Luego, un número de operadores genéticos son aplicados a los padres para producir nuevos individuos que formarán parte de la nueva población.

Cada individuo es representado directamente con el conjunto de los enlaces que conforman el árbol multicast [1, 8].

Siguiendo el esquema del SPEA [3], el algoritmo mantiene una población evolutiva P y un conjunto externo P_{nd} de soluciones óptimas. Empezando con una población aleatoria inicial, los individuos evolucionan a soluciones óptimas, las cuales son almacenadas en el conjunto externo. La Figura 2 muestra el pseudocódigo del MMA2.

```

- Leer grupo multicast a enrutar.
- Inicializar población P.
Hacer {
  - Reemplazar individuos iguales de P.
  - Evaluar Individuos de P.
  - Marcar soluciones no dominadas de P.
  - Actualizar el conjunto de soluciones no dominadas  $P_{nd}$ 
  - Calcular la adaptabilidad de los individuos de P y  $P_{nd}$ 
  - Seleccionar individuos del conjunto  $P \cup P_{nd}$ 
  - Aplicar los operadores genéticos de cruzamiento y mutación.
} mientras el criterio de parada no sea alcanzado.

```

Figura 2. Algoritmo MMA2 propuesto.

```

PrimRST( $G(V,E), s, N$ ) //recibe la red  $G(V, E)$  y el grupo  $s, N$ .
 $T_p = \{\emptyset\}$ ; //Inicialización del árbol a crear (vacío)
 $V_c = \{s\}$ ; //Nodos conectados a  $T_p$ 
 $A = \{(s, j) \in E \mid j \in V\}$ ; //Enlaces candidatos a ser añadidos a  $T_p$ 
Mientras ( $N \cup \{s\} \not\subseteq V_c$ )
  Elegir aleatoriamente un enlace  $(i, j) \in A; i \in V_c$ ;
   $A = A - \{(i, j)\}$ ;
  Si  $j \notin V_c$  entonces // conectar  $j$  al árbol  $T_p$ 
     $T_p = T_p \cup \{(i, j)\}$ 
     $V_c = V_c \cup \{j\}$ ;
     $A = A \cup \{(j, w) \in E \mid w \notin V_c\}$ ;
  Fin si
Fin Mientras
Borrar los enlaces inútiles de  $T_p$ 
Retornar  $T_p$ 

```

Figura 3. Procedimiento utilizado en MMA 2 para la construcción de un árbol T_p .

Leer grupo multicast a enrutar. Se lee el conjunto $\{s, N\}$ de nodos que conforman el grupo multicast a enrutar.

Inicializar P. Este procedimiento genera $|P|$ individuos aleatoriamente, donde P es la población. MMA 2 propone un algoritmo de construcción de árboles basado en el algoritmo de Prim [13]. Empezando con el nodo fuente s , el algoritmo expande el árbol eligiendo un nuevo enlace en cada iteración, añadiendo de esta forma un nuevo nodo al árbol. La elección del enlace se hace aleatoriamente. De esta forma, el algoritmo, denominado PrimRST (*Prim Random Steiner Tree*), inicializa la población evolutiva P . La población P_{nd} es inicializada con aquellos individuos no dominados pertenecientes a la primera generación de P . El pseudocódigo del procedimiento PrimRST es mostrado en la Figura 3.

Descartar individuos iguales en P. En P , puede haber dos o más individuos iguales, lo cual puede reducir la capacidad de búsqueda del algoritmo. Hwang et al. [10] propusieron reemplazar los individuos repetidos por otros nuevos creados aleatoriamente. Dicho enfoque es utilizado en la presente propuesta.

Evaluar P. Este procedimiento calcula el vector objetivo de cada individuo de P , usando las funciones objetivo definidas en las ecuaciones (4.1), (4.2), (4.3) y (4.4).

SPEA. Los siguientes procedimientos son parte del SPEA [3]:

- **Marcar las soluciones no dominadas de P:** analiza la población en cada generación y marca aquellos individuos de P que no son dominados por ningún otro individuo de la población, es decir, las mejores soluciones.
- **Actualizar el conjunto externo de soluciones no dominadas P_{nd} :** cada individuo marcado de P es comparado con los individuos de P_{nd} . Si un individuo marcado es no dominado

respecto a P_{nd} , entonces es copiado a este conjunto. Además, todos los individuos P_{nd} dominados por el individuo marcado son removidos del conjunto de soluciones óptimas.

- **Calcular adaptabilidad de los individuos de P y P_{nd}** : El procedimiento consta de dos pasos:

1. para cada $i \in P_{nd}$ es calculado la fuerza $q_i \in [0,1)$ del individuo, el cual es proporcional al

$$\text{número de individuos } j \in P \text{ tal que } i \triangleright j : q_i = \frac{|j | j \in P_{nd} \wedge i \triangleright j|}{|P|}.$$

2- para cada $j \in P$ es calculado la fuerza $q_j \in [1, N)$ sumando la fuerza de todos los individuos

$$i \in P_{nd} \text{ tal que } i \triangleright j, \text{ más uno: } q_j = 1 + \sum_{i \in P_{nd}, i \triangleright j} q_i.$$

La adaptabilidad (o *fitness*) podría ser entonces calculada como la inversa de la fuerza, pero no es necesario su cálculo explícito por el tipo de operador de selección implementado.

Selección. El operador de selección es aplicado en cada generación sobre el conjunto $P_{nd} \cup P$ con el objeto de seleccionar buenos individuos que generarán la siguiente población P . En MMA2 se propone la utilización del torneo binario [6] como operador de selección. Este operador selecciona pares de individuos, de los cuales aquel individuo de menor fuerza gana el torneo y es seleccionado para la siguiente generación.

Cruzamiento y mutación. El operador de cruzamiento de dos puntos es aplicado sobre cada par de individuos seleccionados. Este operador representa un mecanismo de búsqueda local que enfoca la atención en aquellos individuos con valores altos de adaptabilidad, explorando la información disponible de los padres [6]. Por otra parte, la mutación consiste en cambiar con probabilidad P_{mut} algunos genes en cada cromosoma de la nueva población. El operador de mutación provee un mecanismo de exploración de nuevas zonas del espacio de soluciones [6] y consiste en:

1- identificar los enlaces comunes de ambos padres y copiarlos al individuo hijo. De acuerdo al procedimiento de selección explicado, los individuos con menor fuerza tienen mayor probabilidad de ser seleccionados. Por lo tanto, los enlaces de los padres posiblemente sean “buenos”. Sin embargo, considerar solamente estos enlaces puede conducir a un individuo hijo que consiste en sub-árboles separados, y por lo tanto algunos nuevos enlaces deben ser añadidos;

2- conectar los sub-árboles hasta formar un árbol multicast. En esta etapa, los sub-árboles son conectados de forma aleatoria, y cada sub-árbol tiene asignado un nodo raíz. El algoritmo de interconexión añade en cada iteración un enlace cuyo nodo origen forma parte de un sub-árbol. Dos sub-árboles son conectados cuando el enlace elegido tiene como nodo destino la raíz de uno de los sub-árboles – T_1 – y como origen un nodo perteneciente al otro sub-árbol – T_2 –. La raíz del nuevo sub-árbol (compuesto por los dos sub-árboles) es el nodo raíz de T_2 . El algoritmo termina cuando todo el grupo multicast es conectado. Luego, los enlaces no utilizados en la interconexión del grupo son borrados del árbol. La Figura 4 ilustra un ejemplo de cruzamiento entre dos individuos seleccionados.

Por último, es necesario apuntar que el operador de mutación no fue considerado en MMA 2, debido a que el procedimiento que descarta individuos iguales provee de suficiente diversidad genética a la población.

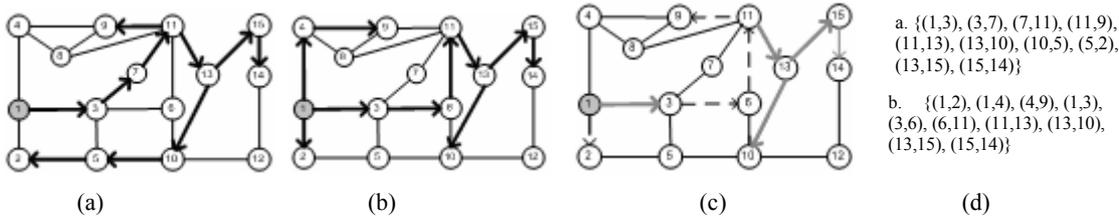


Figura 4. (a), (b) Individuos seleccionados, sobre los cuales se realiza la operación de cruzamiento. (c) Resultado del operador de cruzamiento. Los enlaces discontinuos son seleccionados aleatoriamente para unir los sub-árboles, mientras que los enlaces continuos pertenecen a ambos padres y son heredados por el nuevo individuo. (d) Representación de los árboles de las figuras (a) y (b)

6 Resultados Experimentales

En esta Sección se presentan los experimentos y las simulaciones realizadas con el objetivo de evaluar MMA2.

En la sub-sección 6.1 se presentan los resultados al comparar MMA2 con MMA1, en un problema de prueba estático. En este problema, se conoce el frente Pareto óptimo. Por lo tanto, se evalúan los algoritmos considerando como métricas de desempeño la cantidad de soluciones óptimas encontradas y el tiempo de cómputo. Posteriormente, en la sub-sección 6.2, los algoritmos MMA1, MMA2 y SK son evaluados utilizando problemas de prueba dinámicos, en los cuales se simula el comportamiento de una red de computadoras donde las solicitudes multicast llegan a la red una tras otra.

6.1 Problema de Prueba Estático

Este problema de prueba se realizó en la topología de la red de la NSF (*National Science Foundation*), mostrada en la Figura 5. Cada enlace (i, j) tiene asociado un costo c_{ij} y un retardo d_{ij} en ms, en este orden.

Para el grupo multicast considerado $s = 5$ y $N = \{0, 4, 9, 10, 13\}$, el conjunto P_{true} fue hallado por búsqueda exhaustiva y tiene 16 soluciones Pareto óptimas. El tiempo de corrida fue de aproximadamente 3 horas [14]. En la Figura 6 se grafica este conjunto en un espacio de tres dimensiones (sin considerar α_T), donde cada solución tiene asignada una letra.

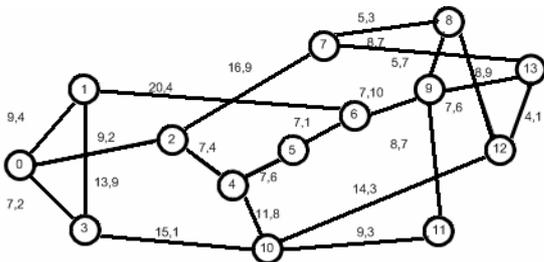


Figura 5. Red de la NSF

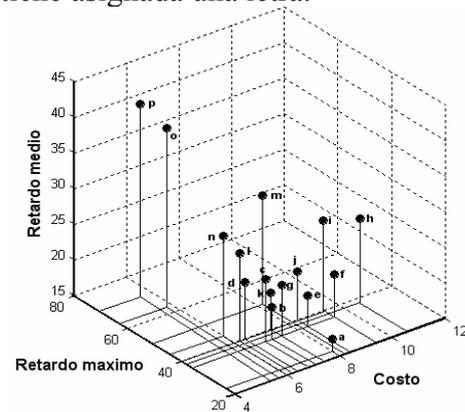


Figura 6. Ubicación de Y_{true} en \mathbb{R}^3 , sin considerar α_T

Para este problema, se hicieron 100 corridas de cada algoritmo. Para MMA1 [14], $|P| = 40$, $R = 30$ (el cual define la cantidad de rutas de menor retardo, costo y utilización máxima de los enlaces, necesarias para la inicialización de la población) y $P_{mut}=0.4$. Para MMA2, se estableció $|P| = 25$. Las corridas pararon al alcanzar 100 ms.

Las métricas de comparación de este problema fueron: número de soluciones Pareto óptimas máxima (S_{max}), mínima (S_{min}), promedio (S_a) obtenidas en las corridas, y la desviación estándar (σ_s) de las mismas. Los resultados son presentados en la Tabla 1.

La Tabla 1 muestra que ambos algoritmos tuvieron un buen rendimiento es este problema de prueba, encontrando al menos 14 soluciones óptimas (87,5 % del conjunto Pareto).

Tabla 1. Resultados del Problema 2

	MMA2	MMA1
S_{max}	16	15
S_{min}	14	14
S_g	15.82	14.13
σ_s	0.41	0.1

MMA1 no encontró en ninguna oportunidad, 15 soluciones óptimas en 2 corridas y 14 en 98 oportunidades. Por su parte, MMA2 encontró P_{true} en 83 ocasiones, 15 soluciones óptimas en 16 corridas y 14 en una oportunidad. Claramente, MMA2 tuvo un mejor rendimiento que MMA1. Note que, en el 98 % de las corridas, MMA1 halló 14 soluciones. En dichas corridas, las dos soluciones que MMA1 no ha podido encontrar fueron los árboles (o) y (p) que están situados en una esquina del espacio objetivo del frente Pareto. (Ver Figura 6)

Dado que ambos algoritmos están basados en el SPEA, y MMA2 no ha tenido el problema citado arriba, es importante apuntar las diferencias entre ambos algoritmos, para comprender la diferencia de rendimiento. Como se explicó en el Capítulo 4, la principal diferencia entre MMA1 y MMA2 es la representación cromosómica. De esto se deriva que el cruzamiento también difiere. Por lo tanto, se infiere que la codificación utilizada en MMA2 da a este algoritmo una mayor capacidad de exploración en el dominio X_f . Esta afirmación será confirmada en los problemas de prueba dinámicos.

6.2 Problemas de Prueba Dinámicos

MMA1 y MMA2 también fueron evaluados en situaciones dinámicas utilizando la red de la NTT[14] que consta de 55 nodos y 144 enlaces de 6 Mbps. Para estas pruebas, $\forall (i,j) \in E, c_{ij} = 1$. Formulando de esta manera, C_T (ecuación 4.2) calcula el consumo total de ancho de banda.

Dos corridas distintas fueron realizadas. La primera corrida consiste en solicitudes multicast en las cuales el tamaño de los grupos varía entre 3 y el 35 % del número total de nodos de la red. En la segunda corrida, el tamaño de los grupos varía entre 35% y 70% del número de nodos de la red. De esta forma, los algoritmos propuestos se evalúan en modos denso y esparzo [2]. Para cada caso, 400 solicitudes multicast fueron creadas aleatoriamente. Cada solicitud consiste en un nodo origen s , un conjunto de nodos destinos N y una demanda de tráfico $\phi = 400$ Kbps. La duración de cada solicitud de tráfico tuvo una distribución exponencial con promedio de 60 segundos. El instante de llegada de cada solicitud fue aleatoriamente distribuida entre 0 y 1800 segundos. El parámetro R de MMA1 fue $R = 30$ y $P_{mut} = 0.4$.

Dado que MMA1 y MMA2 proveen más de una solución óptima para cada grupo, el árbol de menor α_T fue elegido. En caso de existir más de un árbol con α_T mínimo, aquel árbol con menor C_T fue seleccionado. Es decir, el de menor consumo de ancho de banda.

MMA1 y MMA2 fueron comparados entre sí, y con el algoritmo propuesto por Seok et al. [2]. Dado que SK tiene asociado el parámetro H definido a priori, en cada corrida se ha variado dicho parámetro, y las distintas soluciones proveídas de esta manera se han comparado con MMA1 y MMA2. Para más detalles del algoritmo SK, se deriva al lector a [2].

De forma a evaluar cuantitativamente los resultados obtenidos en estas simulaciones dinámicas, se han considerado las métricas de dominancia definidas a continuación:

ND_{A1-A2}: Número de árboles construidos por el algoritmo 1 (A_1) que dominan a los correspondientes árboles construidos por el algoritmo 2 (A_2).

ND_{A2-A1}: Número de árboles construidos por el algoritmo 2 (A_2) que dominan a los correspondientes árboles construidos por el algoritmo 1 (A_1).

I_{A1-A2} : Número de árboles construidos por el algoritmo 1 (A_1) indiferentes (no pueden considerarse mejor ni peor) que los correspondientes árboles construidos por el algoritmo 2 (A_2).

Además, con el objeto de mostrar cualitativamente los resultados, se han graficado los valores normalizados a MMA2 de la utilización máxima de los enlaces de la red, el consumo total de ancho de banda y el retardo total, de las simulaciones hechas sobre la red NTT. Los valores normalizados se definen a continuación:

-Utilización máxima de los enlaces, normalizada a MMA2: $\alpha_N^W = (\alpha_w - \alpha_{MMA2}) / \alpha_{MMA2}$ (6.1)

-Consumo total de ancho de banda, normalizado a MMA2: $B_N^W = (B_w - B_{MMA2}) / B_{MMA2}$ (6.2)

-Retardo total, normalizado a MMA2: $D_N^W = (D_w - D_{MMA2}) / D_{MMA2}$ (6.3)

donde

w : algoritmo a comparar con MMA2.

α_w : utilización máxima de los enlaces de la red utilizando w .

α_{MMA2} : utilización máxima de los enlaces de la red utilizando MMA2.

B_w : consumo total de ancho de banda de la red utilizando w .

B_{MMA2} : consumo total de ancho de banda utilizando MMA2.

D_w : retardo total usando w , dado por la suma del retardo total de los grupos multicast en la red. El retardo total de un grupo está dado por la suma de los retardos de extremo a extremo a cada uno de los nodos destinos.

D_{MMA2} : retardo total usando MMA2.

6.2.1 Corrida 1. Grupos multicast pequeños/ medianos. Modo Esparzo

En esta corrida, todas las solicitudes de tráfico fueron enrutadas. Es decir, no hubo una sola solicitud multicast rechazada por falta de capacidad de la red. Los parámetros utilizados se listan en la Tabla 2. La Tabla 3 muestra los resultados obtenidos.

De la Tabla 3 se puede concluir que el algoritmo de mejor rendimiento, considerando la métrica de dominancia, fue MMA2, dado que construyó mejores árboles que MMA1 y SK. De las 400 solicitudes multicast, 89 árboles de MMA2 dominaron a los correspondientes árboles de MMA1, mientras que MMA1 solo construyó 51 árboles mejores que los de MMA2. Además, en el caso más favorable al algoritmo SK, cuando $H = 0$, 101 árboles construidos por MMA2 dominaron a los respectivos árboles de SK ($H = 0$). De forma similar, MMA1 construyó 141 árboles mejores que SK ($H = 0$). Note que si se considera SK ($H = 6$), la diferencia a favor de MMA1 y MMA2 fue aún mayor. Ambos algoritmos dominaron a SK ($H = 6$) en más de 300 casos.

Los tiempos medio de cómputo de un árbol multicast de SK fueron: 15 ms ($H = 0$), 25 ms ($H = 3$) y 60 ms ($H = 6$). Por su parte, los tiempos de MMA1 y MMA2 fueron 325 ms y 465 ms, respectivamente. Es necesario aclarar que, si bien los tiempos de MMA1 y MMA2 son mayores, estos algoritmos proveen un conjunto amplio de soluciones, del cual puede ser elegida la solución más adecuada para cada ocasión. En estas pruebas dinámicas simplemente se toma una de las soluciones óptimas, de forma a poder comparar ambos algoritmos con un mono-objetivo como SK.

Tabla 2. Parámetros de las corridas

		P	Gen.	T.Grupo
C1	MMA1	50	500	[3, 20]
	MMA2	40	60	
C2	MMA1	60	500	[20, 40]
	MMA2	50	80	

Tabla 3. Resultados de la corrida 1

A1 - A2	ND _{A1-A2}	%	ND _{A2-A1}	%	I_{A1-A2}	%
MMA1 - MMA2	51	12,75	89	22,25	260	65
MMA1 - SK (H=0)	141	35,25	8	2	251	62,75
MMA1 - SK (H=3)	239	59,75	5	1,25	156	39
MMA1 - SK (H=6)	303	75,75	6	1,5	91	22,75
MMA2 - SK (H=0)	101	25,25	0	0	299	74,75
MMA2 - SK (H=3)	216	54	2	0,5	182	45,5
MMA2 - SK (H=6)	315	78,75	1	0,25	84	21

Las Figuras 7(a) a 7(d) muestran los valores normalizados de la utilización máxima de los enlaces de la red, dados por las Ecuaciones 6.1 a 6.3. Note que la utilización máxima de los enlaces de SK ($H = 0$) fue en casi todo momento superior a la utilización máxima de los enlaces de MMA2. En gran parte del tiempo, $\alpha_N^{SK(H=0)}$ toma valores entre 0.2 y 0.6, indicando que SK ($H = 0$) produce una utilización máxima al menos 20% mayor que la utilización máxima de MMA2. Con $H = 3$ y $H = 6$, SK mejoró esta métrica. Aún así, $\alpha_N^{SK(H=3)}$ y $\alpha_N^{SK(H=6)}$ son positivos en gran parte del tiempo. $\alpha_N^{SK(H=3)}$ varía entre -0,3 y 0.5, mientras que $\alpha_N^{SK(H=6)}$ toma valores entre -0,35 y 0.5. Por su parte, MMA1 produjo una utilización máxima de los enlaces normalizada con valores que varían entre -0,5 y 0,35, mostrando que minimizó esta métrica en forma similar a MMA2.

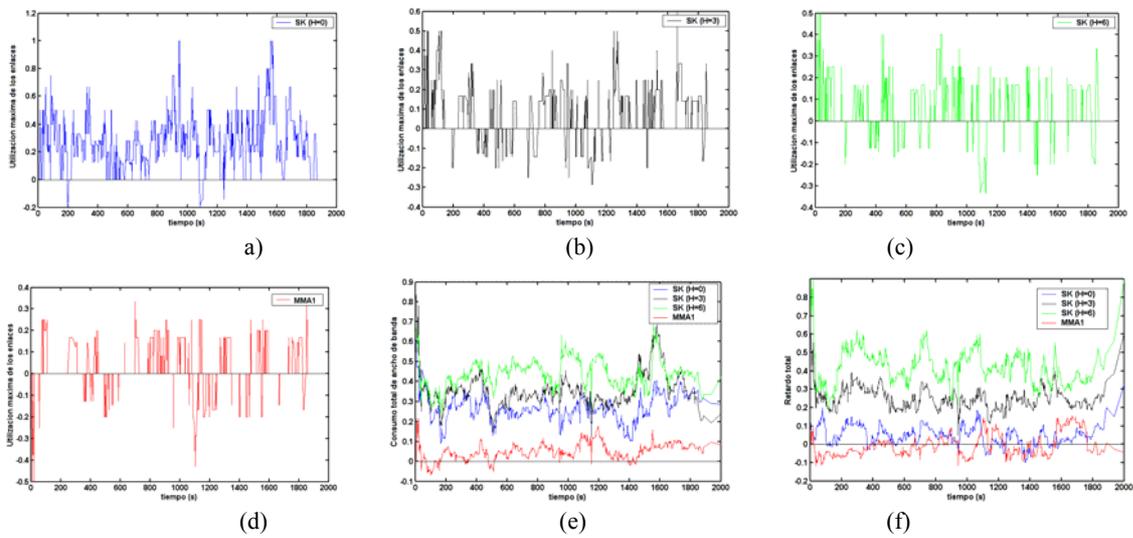


Figura 7. Valores normalizados a MMA2 de la corrida 1. (a) $\alpha_N^{SK(H=0)}$. (b) $\alpha_N^{SK(H=3)}$. (c) $\alpha_N^{SK(H=6)}$. (d) α_N^{MMA1} . (e) Consumo total de ancho de banda. (f) Retardo total

La Figura 7(e) muestra el consumo total de ancho de banda normalizado a MMA2, dado por la Ecuación 6.2. Note que todas las curvas tienen valores positivos, indicando que MMA2 consumió una menor cantidad de ancho de banda que SK y MMA1. Al aumentar H , SK aumentó el consumo de ancho de banda. Con $H = 0$ y $H = 3$, SK utilizó aproximadamente entre 20 y 45% más ancho de banda que MMA2, mientras que con $H = 6$ la utilización de recursos fue aún mayor. Note que, en $t \approx 1600$ segundos, la cantidad de ancho de banda utilizada con $H = 3$ y $H = 6$ alcanzó el pico máximo y $B_N^{SK(H=3)} \approx B_N^{SK(H=6)} \approx 0.7$. Por su parte, la curva de consumo de MMA1 también es mayor que 0 en casi todo momento. A pesar de ello, claramente MMA1 también tiene un consumo menor que SK.

La Figura 7(f) muestra el retardo total normalizado a MMA2, dado por la Ecuación 6.3. D_N^{MMA1} toma valores entre -0.1 y 0.1, indicando que, por momentos, el retardo de MMA2 fue menor que el de MMA1 y viceversa. Por otra parte, SK produjo retardos notoriamente superiores a los de MMA2 y MMA1. La mayor parte del tiempo $D_N^{SK(K=0)}$ toma valores positivos entre 0 y 0.15. Al aumentar H , SK incrementó el retardo total, y la diferencia entre los algoritmos propuestos en este trabajo y SK fue aún mayor. Por ejemplo, cuando $H = 6$, el retardo de SK alcanzó picos muy elevados, aproximándose el retardo total a un valor 60% mayor que el retardo de MMA2 en $t \approx 300$, $t \approx 750$ y $t \approx 1100$.

6.1.2 Corrida 2. Grupos multicast grandes. Modo Denso.

En esta corrida, algunas solicitudes fueron rechazadas por falta de capacidad de la red. La Tabla 4 muestra el número de solicitudes aceptadas por cada algoritmo. Considerando el número de solicitudes aceptadas, MMA2 obtuvo el mejor resultado. Sin embargo, la diferencia entre todos los algoritmos fue mínima.

La Tabla 5 muestra las relaciones de dominancia entre los algoritmos comparados. Note que MMA1 construyó 26 árboles que dominaron a MMA2, mientras que 40 árboles construidos por MMA2 dominaron a los correspondientes árboles de MMA1. Comparado con SK, MMA1 generó 180 ($H = 0$), 231 ($H = 3$) y 254 ($H = 6$) árboles mejores que los árboles construidos por este algoritmo. Es decir, para aproximadamente el 50% de las solicitudes multicast, los árboles de MMA1 dominaron a SK. La relación de dominancia entre MMA2 y SK nuevamente favorece al algoritmo propuesto en este trabajo: 88 ($H = 0$), 205 ($H = 3$) y 251 ($H = 6$) árboles construidos por MMA2 fueron mejores que los árboles construidos por SK. En contrapartida, 9 ($H = 0$), 5 ($H = 3$) y 5 ($H = 6$) árboles de SK dominaron a los respectivos árboles de MMA2.

Tabla 4. Solicitudes aceptadas

	S. acept.	%
MMA1	336	84
MMA2	340	85
SK ($H=0$)	336	84
SK ($H=3$)	339	84,75
SK ($H=6$)	339	84,75

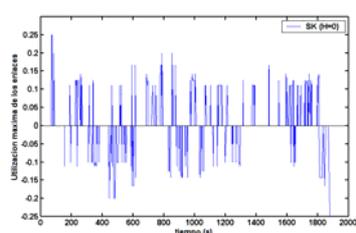
Tabla 5. Resultados de la corrida 2

A1 - A2	ND _{A1-A2}	%	ND _{A2-A1}	%	I _{A1-A2}	%
MMA1 - MMA2	26	6,5	40	10	334	83,5
MMA1 - SK ($H=0$)	180	45	11	2,75	209	52,25
MMA1 - SK ($H=3$)	231	57,75	8	2	161	40,25
MMA1 - SK ($H=6$)	254	63,5	8	2	138	34,5
MMA2 - SK ($H=0$)	88	22	9	2,25	303	75,75
MMA2 - SK ($H=3$)	205	51,25	5	1,25	190	47,5
MMA2 - SK ($H=6$)	251	62,75	5	1,25	144	36

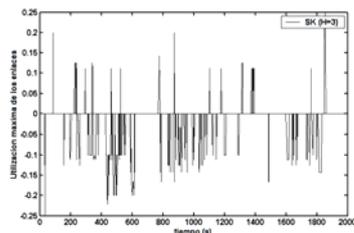
Las Figuras 8(a) a 8(d) muestran los valores normalizados de la utilización máxima de los enlaces. Considerando esta métrica, MMA2, MMA1 y SK demuestran un rendimiento similar, y de las gráficas no se puede concluir que un algoritmo produce una utilización máxima de los enlaces menor que otro, pues en algunos intervalos de tiempo, un algoritmo gana al otro y viceversa.

La Figura 8(e) muestra el consumo total de ancho de banda normalizado a MMA2. Nuevamente en esta corrida, MMA2 claramente consume una menor cantidad de ancho de banda para enrutar las solicitudes multicast que MMA1 y SK. Note además que MMA1 también consume menos ancho de banda que SK. Mientras MMA1 consume entre 0 y 10% más ancho de banda que MMA2 en la mayor parte del tiempo, SK consume entre 20 y 40% más. Al Aumentar H , SK aumenta el consumo y la diferencia entre los algoritmos propuestos en este trabajo y SK es aún mayor.

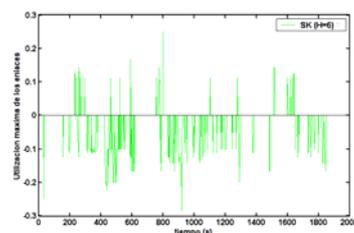
La Figura 8(f) muestra el retardo total normalizado a MMA2. Como en la corrida 1, MMA1 produce un retardo total menor que los demás algoritmos. El retardo total de este algoritmo varía entre 0 y 20% menos que al retardo total de MMA2. También se puede observar que el retardo total de MMA2 es menor que el retardo de SK con sus diferentes parámetros de H . Por ejemplo, con $H = 6$, el retardo de SK en casi todo momento es, como mínimo, 40% mayor que el retardo de MMA2.



(a)



(b)



(c)

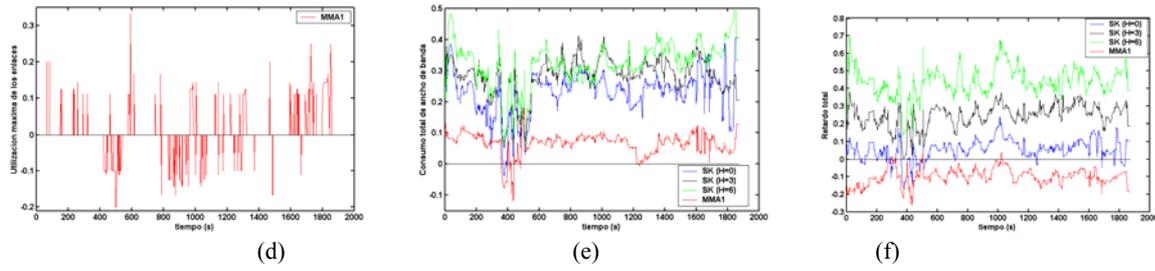


Figura 8. Valores normalizados a MMA2 de la corrida 2. (a) $\alpha_N^{SK(H=0)}$. (b) $\alpha_N^{SK(H=3)}$. (c) $\alpha_N^{SK(H=6)}$. (d) α_N^{MMA1} . (e) Consumo total de ancho de banda. (f) Retardo total

Las simulaciones dinámicas han demostrado que los algoritmos propuestos en este trabajo han obtenido resultados satisfactorios en ambos modos: denso y esparzo. MMA2 construyó un mayor número de árboles multicast que dominaron a las correspondientes soluciones de MMA1 y SK. Además, en modo esparzo, ambos algoritmos construyeron árboles que dominaron a los respectivos árboles de SK ($H=0$). Con valores mayores de H , la tasa de dominancia a favor de MMA1 y MMA2 fue aún mayor al 50 %. Cifras similares se obtuvieron para la corrida en modo denso, confirmando las ventajas de utilizar el algoritmo MMA2 aquí propuesto.

Si bien el número de solicitudes multicast enrutadas por los tres algoritmos fue similar, MMA2 enrutó la mayor cantidad de solicitudes en modo denso, consumiendo una menor cantidad de ancho de banda que MMA1 y SK y obteniendo un retardo total menor que SK.

7 Conclusiones y Trabajos Futuros

El problema de enrutamiento multicast en redes de computadoras, con más de una métrica a considerar, generalmente es tratado como un problema de optimización mono-objetivo, donde las funciones objetivo son combinadas escalarmente a través de una suma ponderada. El presente trabajo trata este problema como puramente multiobjetivo, donde la utilización máxima de los enlaces del árbol α_T , el costo C_T , el retardo máximo de extremo a extremo D_M y el retardo medio D_A son tratados como funciones objetivo a minimizar en forma simultánea e independiente.

Para resolver este problema, dos nuevos algoritmos evolutivos basados en el SPEA fueron estudiados: MMA1 y MMA2. Estos algoritmos obtienen un conjunto Pareto óptimo de soluciones en una corrida. Esta característica es de gran importancia, pues la solución más adecuada puede ser elegida para cada caso particular, sin considerar restricciones a priori.

Dado que hasta la fecha no han sido suficientemente publicados trabajos sobre enrutamiento multicast multiobjetivo ni *test beds* con problemas de prueba que sirvan de base comparativa y experimental, los algoritmos propuestos en este trabajo fueron evaluados con pequeños problemas de prueba estáticos y dinámicos formulados para la ocasión. El análisis de los resultados experimentales muestra que la representación cromosómica de MMA2 provee a este algoritmo una mayor capacidad de exploración, permitiendo obtener aquellas soluciones Pareto óptimas ubicadas en “esquinas” del espacio de búsqueda que no siempre son halladas por MMA1.

En situaciones dinámicas, las simulaciones mostraron el buen funcionamiento del algoritmo propuesto en este trabajo. En la mayoría de los casos, en ambos modos (denso y esparzo), los árboles proveídos por MMA1 y MMA2 dominaron a los árboles de SK. Además, MMA2 obtuvo una ligera mejora de rendimiento que MMA1 en cuanto a dominancia de soluciones se refiere. Es decir, tuvo al menos una función objetivo menor, y las demás funciones objetivo menor o igual que las funciones objetivo de los árboles hallados por SK y MMA1. De esta forma, no solo se obtienen soluciones con valores de retardo medio y máximo menores, sino también se minimiza la utilización de los enlaces y el costo total en términos de recursos, como ancho de banda consumido.

Como trabajos futuros, los siguientes tópicos son propuestos: evaluación comparativa de MMA1, MMA2 y nuevos algoritmos multiobjetivos basados en otros MOEAs como SPEA2, NSGA, NSGA2 y CNSGA2; estudio e implementación de nuevos esquemas de ingeniería de tráfico multicast multi-árboles, donde el flujo de datos de un grupo multicast es transmitido a los nodos destinos a través de varios árboles; definición de un *test bed* con problemas de prueba que sirvan de base comparativa y experimental para los algoritmos desarrollados y publicados, extendiendo los paradigmas mono-objetivo a multiobjetivos, tal como se hizo con el Problema de Prueba 1.

Agradecimiento

Se agradece a la Universidad Católica “Nuestra Señora de la Asunción”, por su apoyo al proyecto “Ingeniería de Tráfico utilizando optimización multi-objetivo” que sirve de marco a la realización del presente trabajo.

Referencias

- [1] C. Ravikumar, y R. Bajpai, “Source-based delay bounded multicasting in multimedia networks”, *Computer Communications*, Vol. 21, 1998, pág. 126-132.
- [2] Y. Seok, Y. Lee, Y. Choi, y C. Kim, “Explicit Multicast Routing Algorithm for Constrained Traffic Engineering”, *IEEE Seventh International Symposium on Computer and Communications (ISCC'02)*, Italia, 2002.
- [3] 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, pp. 257-271.
- [4] Y. Donoso, R. Fabregat, y J. Marzo, “Multi-objective Optimization Algorithm for Multicast Routing with Traffic Engineering”, *IEEE 3^{er} International Conference on Networking (ICN'2004)*, Guadalupe, Caribe Francés, 2004.
- [5] V. Kompella, J. Pasquale y G. Polyzos, “Multicast routing in multimedia communication”, *IEEE/ACM Transactions on Networking*, Vol. 1 No. 3, 1993, pág. 286-291.
- [6] D. Goldberg, *Genetic Algorithm is Search, Optimization & Machine Learning*, Addison Wesley, 1989.
- [7] W. Zhengying, S. Bingxin, y Z. Erdun, “Bandwidth-delay-constraint least-cost multicast routing based on heuristic genetic algorithm”, *Computer Communications*, Vol. 24, 2001, pág. 685-692.
- [8] P. de Araujo, y G. Barbosa, “Multicast Routing with Quality of Service and Traffic Engineering Requirements in the Internet, Based On Genetic Algorithm”, *Proceedings of the VII Brazilian Symposium on Neural Networks (SBRN'02)*, 2002.
- [9] J. Yen, “Finding the k shortest loopless path in a network”, *Management Science*, 17:712-716, 1971.
- [10] R.H. Hwang, W.Y. Do, y S.C. Yang, “Multicast Routing Based on Genetic Algorithms”, *Journal of Information Science and Engineering*, Vol. 16, 2000, pág. 885-901.
- [11] B. Barán, y R. Sosa, “A New Approach for Antnet Routing”, *IEEE International Conference on Computer Communication and Networks (ICCCN'2000)*, USA, 2000.
- [12] N. Spring, R. Mahajan, y D. Wetherall. “Measuring ISP topologies with Rocketfuel”, *Proceedings of the ACM SIGCOMM'02 Conference*, ACM, 2002.
- [13] E. Horowitz, y S. Sahni, *Fundamentos de Estructura de Datos*, Campus LTDA, 1986.
- [14] J. Crichigno, y B. Barán, “Multiobjective Multicast Routing Algorithm”, *IEEE 11th International Conference on Telecommunications*, Brasil, 2004.
- [15] J. Crichigno, y B. Barán, “A Multicast Routing Algorithm Using Multiobjective Optimization”, *IEEE 11th International Conference on Telecommunications*, Brasil, 2004.