

Aplicación de NSGA-II y SPEA-II para la optimización multiobjetivo de redes multicast*

Carolina Alvarado**, Iván Herazo **, Carlos Ardila***,
Yezid Donoso****

Departamento de Ingeniería de Sistemas y Computación,
Universidad del Norte, Barranquilla (Colombia)

Resumen

En este artículo se aplican los algoritmos evolutivos para optimización multiobjetivo, Non-dominated Sorting Genetic Algorithm (NSGA-II) y Strength Pareto Evolutionary Algorithm (SPEA-II). Para esto se toma como referencia un problema de optimización en una red de datos multicast, el cual tiene como funciones objetivo el número de saltos y el retardo en la transmisión. El rendimiento de los algoritmos se compara en tres topologías de red de tamaños diferentes. Además, el modelo es resuelto para dos de las topologías utilizando la herramienta GAMS, y los resultados se comparan con las soluciones obtenidas mediante los algoritmos propuestos. Los resultados del problema muestran el rendimiento de los algoritmos en la solución del mismo.

Palabras claves: Optimización multiobjetivo, algoritmos evolutivos, multicast.

Abstract

In this paper, an analysis of evolutionary algorithms for multi objective optimization, Non-dominated Sorting Genetic Algorithm (NSGA-II) and Strength Pareto Evolutionary Algorithm (SPEA-II) is presented. For this analysis, is taken as reference an optimization problem in a multicast data network, which has as objective functions the hop count and transmission delay. The algorithms performance is compared in tree different networks. Moreover, the model for two of this topologies using GAMS tool is resolved and results are compared with the NSGA-II and SPEA-II algorithms proposed. Problem results show the algorithms performance in their solution.

Key words: Multiobjective optimization, evolutionary algorithms, multicast.

Fecha de recepción: 6 de diciembre de 2004
Fecha de aceptación: 25 de mayo de 2005

* Este artículo forma parte de los resultados de la investigación "Aplicación de Ingeniería de Tráfico en Redes Multicast".

** Ingenieros de sistemas, Universidad del Norte. calvarado@unimail.uninorte.edu.co y iherazo@unimail.uninorte.edu.co

*** Ingeniero de sistemas, Universidad del Norte. Especialista en Matemáticas. Estudiante maestría en Ingeniería Industrial, Universidad del Norte. cardila@uninorte.edu.co

**** Ingeniero de sistemas, Universidad del Norte. Magíster en Ingeniería de Sistemas y Computación, Universidad de los Andes. Ph.D. en Redes Telemáticas, Universidad de Girona (España). Profesor del Departamento de Ingeniería de Sistemas, Universidad del Norte. Dirección postal: A.A. 1569, Barranquilla, Colombia. ydonoso@uninorte.edu.co

1. INTRODUCCIÓN

Las nuevas aplicaciones que están surgiendo en Internet han generado un aumento en la necesidad de transmitir información desde un mismo origen a múltiples destinos (multicast) y además, que en esta transmisión se garanticen ciertos parámetros de *Calidad del Servicio* (QoS, por sus siglas en inglés), como pueden ser, entre otros, el retardo máximo de los paquetes, el costo y el número de paquetes que pueden ser descartados sin afectar la calidad en la transmisión de la información.

Por su parte, los Algoritmos Evolutivos Multiobjetivo (MOEAs) combinan técnicas de computación evolutiva con la teoría de optimización multiobjetivo, y por lo tanto brindan la posibilidad de realizar búsquedas en espacios ilimitados y complejos. Igualmente, permiten mantener toda una población de soluciones óptimas, por lo que se han planteado como una buena herramienta algorítmica para resolver diversos problemas de optimización de objetivos múltiples. Algunos de los algoritmos evolutivos más conocidos son NSGA, SPEA, NSGA-II, SPEA-II y PAES-II, entre otros.

En este artículo se aplican NSGA-II y SPEA-II en la minimización del número de saltos y el retardo de transmisión en una red multicast. El objetivo principal es analizar la eficacia y la eficiencia de estos dos algoritmos en la solución del problema propuesto. Para determinar la eficacia de los algoritmos en la solución del problema, también se resuelve el modelo mediante el uso del método de la suma ponderada para dos topologías pequeñas. Las soluciones obtenidas con este método son comparadas con las de los algoritmos propuestos, con el fin de determinar qué tan buena es la aproximación de éstos a las soluciones reales, y cuál de ellos se aproxima mejor al conjunto de soluciones óptimas. Finalmente, se compara el rendimiento de los algoritmos para una topología grande, la cual no se resuelve analíticamente debido a la gran complejidad que esto implica.

El trabajo está organizado de la siguiente manera: En la sección 2 se presentan algunos trabajos relacionados. En la sección 3 se expone la teoría básica relacionada con la optimización multiobjetivo. En la sección 4 se explica la formulación matemática del modelo de optimización propuesto para minimizar el número de saltos y el retardo cuando se transmite tráfico multicast. En la sección 5 se describen los algoritmos evolutivos utilizados para resolver el problema previamente planteado. En la sección 6 se presentan los resultados de los experimentos realizados, y en la sección 7 las conclusiones.

2. TRABAJOS RELACIONADOS CON LA OPTIMIZACIÓN MULTICAST

A continuación se describen varios algoritmos de optimización multicast. Al final se muestra una tabla comparativa para éstos, haciendo énfasis en los objetivos considerados, sus restricciones y los algoritmos genéticos aplicados.

En [2], Banerjee y Das, al igual que Roys y Das en [18], proponen el protocolo de enrutamiento móvil para multitransmisión basado en calidad de servicio (QM^2RP). Como su nombre lo indica, QM^2RP fue pensado para redes multicast móviles con QoS. Los parámetros de calidad de servicio analizados aquí son los siguientes: retardo extremo a extremo, requerimiento de ancho de banda y utilización de ancho de banda residual. Debido a que el algoritmo fue ideado para trabajar sobre redes móviles, los autores lo diseñaron tratando de determinar las rutas para multitransmisión mediante el cumplimiento estocástico (probabilístico) de los parámetros objetivos. El algoritmo utiliza la aproximación de dominancia Pareto en los algoritmos genéticos (AGs) para solucionar el problema multiobjetivo. Un algoritmo similar puede encontrarse en [4], el cual difiere de QM^2RP en la adición de una restricción en la variación del retardo y en la forma como calcula el nivel de aptitud.

En [14], Inagaki *et al.* proponen un algoritmo heurístico para encontrar varios árboles multicast similares. Este algoritmo encuentra varios caminos para un flujo desde un origen hasta un solo destino y no tiene una medida concreta para seleccionar esas rutas.

En [15], Leung *et al.* presentan un algoritmo para encontrar árboles para transmisión de flujo multicast. Sin embargo, este algoritmo no tiene en cuenta restricciones. El algoritmo se basa en (a) la transformación de la red subyacente a su forma completa de distancia, soportada en el peso de los enlaces, y (b) la representación de árboles multicast como una lista de nodos. Por lo tanto, todos los cromosomas son válidos y, en consecuencia, las operaciones de cruce y mutación son más simples. Sin embargo, algunas veces ciertos enlaces que no existen en la topología original deben ser computados como el costo mínimo para un camino entre los vértices que no están conectados, por lo tanto el algoritmo debe utilizar métricas sencillas con pesos escalares.

En [21], Sun propone un algoritmo para encontrar árboles multicast mediante la minimización de una función de costo, tomando el retardo como restricción. Los cromosomas se representan como una cadena binaria de nodos de *Steiner*.

i.e: los nodos que no son ni el origen ni del conjunto de destinos. Similar a [15], la función de costo es una sola medida escalar del objetivo deseado.

En [16], Li muestra una perspectiva diferente con relación a la representación del cromosoma. Aquí, los árboles no son codificados ni decodificados. Esto provoca que los procesos de cruzamiento y mutación trabajen sobre las ramas del árbol mismo. La función objetivo también se mide como un solo valor escalar.

En [22], un Algoritmo Genético (AG) es propuesto con los siguientes objetivos: disponibilidad de ancho de banda, retardo, tasa de pérdida y atenuación extremo a extremo. Este algoritmo también fusiona todas las mediciones en un único objetivo escalar.

Diversos trabajos evalúan el rendimiento de algunos de estos algoritmos evolutivos utilizando métricas de eficacia. En [19], utilizando como referencia 6 funciones test, se presenta una evaluación de NSGA-II, SPEA-II y PESA-II sobre la influencia de los parámetros P_{cruce} , $P_{mutación}$ y la relación entre el tamaño de la población y el tamaño de la población élite. En [12] se analiza el rendimiento de NSGA-II, SPEA-II y NSGA-II con elitismo controlado en diseño de sistemas de seguridad.

Las técnicas algorítmico-genéticas utilizadas en cada uno de los artículos mencionados previamente pueden ser resumidas en la tabla 1:

Tabla 1
Trabajos relacionados y técnicas algorítmico-genéticas utilizadas

| | Codificación/Decodificación | Cruzamiento | Mutación | Nivel de aptitud |
|------|------------------------------|--------------------|------------------------|---------------------------------|
| [2] | Conjunto de nodos usados* | De un punto | Cambio aleatorio común | Función de costo/Suma con pesos |
| [18] | Representación centinela | De un punto | N/A | (MOGA)/Dominancia Pareto |
| [4] | Representación centinela | De un punto | Cambio aleatorio común | (MOGA)/Dominancia Pareto |
| [14] | Codificación de predecesores | De varios puntos | N/A | Función de costo |
| [15] | Conjunto de nodos usados* | De un punto | Cambio aleatorio común | Función de costo/Suma con pesos |
| [21] | Conjunto de nodos usados* | Operaciones And/Or | N/A | Función de costo/Suma con pesos |
| [16] | Vector característico | De árbol | Tree mutation | Función de costo/Suma con pesos |
| [22] | Conjunto de nodos usados* | Operaciones And/Or | N/A | Función de costo/Suma con pesos |

* Estas técnicas de "conjuntos de nodos usados" no son las mismas, porque son específicas del problema tratado.

En [8], [9], [10] y [11], Donoso *et al.* proponen un modelo de optimización para minimizar la máxima utilización de los enlaces, el retardo, el número de saltos y el ancho de banda consumido en una red multicast, igualmente proponen una heurística para resolver el modelo propuesto.

Las características de optimización de los algoritmos descritos arriba pueden ser resumidos en la tabla 2.

Tabla 2
Trabajos relacionados y características de los modelos de optimización considerados

| | OBJETIVOS | | | | | | RESTRICCIONES | | | FLUJO | CAMINOS / ARBOLES | DIVISIÓN |
|---|-----------|----|----|----|----|--|---------------|----|-----|--|-------------------|----------|
| | MLU | HC | DL | BC | LR | JIT | DL | BC | MSF | | | |
| [14], [15], [16]* | | | | | | | | | | Multicast | Solo un árbol | No |
| [2] | | | X | X | | | | | | Multicast | Solo un árbol | No |
| [18] | | | X | X | X | | | | | Multicast | Solo un árbol | No |
| [4] | | | X | X | X | | X | | | Multicast | Solo un árbol | No |
| [21]* | | | | | | | X | | | Multicast | Solo un árbol | No |
| [22] | | | X | X | X | X | | | | Unicast | Solo un árbol | No |
| [8], [9], [10], [11] | X | X | X | X | | | | X | X | Multicast | Múltiples árboles | Sí |
| * Estos algoritmos minimizan (o maximizan) una función de costo. Por lo tanto, estos podrían trabajar con casi todos los objetivos. | | | | | | MLU : Máximo uso del enlace HC : Conteo de saltos DL : Retardo total BC : Ancho de banda total consumido JIT : Atenuación de extremo a extremo | | | | LR: Tasa de pérdida. MSF: Máximo número de subflujos. Manejo de la fracción de flujo por cada nodo egreso a lo largo de un enlace en el modelo de optimización. | | |

3. OPTIMIZACIÓN MULTIOBJETIVO

Un problema de optimización multiobjetivo (*MOP*) es aquel que incluye un conjunto de n variables de decisión, un conjunto de k funciones objetivo, y un conjunto de m restricciones de desigualdad y p restricciones de igualdad, en donde las funciones objetivo y las restricciones son funciones de las n variables de decisión. En su forma general, un *MOP* puede ser expresado matemáticamente como [7]:

$$\begin{aligned}
 \text{Optimizar} & \quad y = f(x) = \{f_1(x), f_2(x), \dots, f_k(x)\} & (3.1) \\
 \text{Sujeto a} & \quad g(x) = \{g_1(x), g_2(x), \dots, g_m(x)\} \geq 0 \\
 & \quad h(x) = \{h_1(x), h_2(x), \dots, h_p(x)\} = 0 \\
 \text{donde} & \quad x = \{x_1, x_2, \dots, x_n\} \in X \\
 & \quad y = \{y_1, y_2, \dots, y_n\} \in Y
 \end{aligned}$$

x representa el vector de decisión, X denota el espacio de decisión, y representa el vector objetivo y el espacio objetivo es denotado por Y . Al conjunto de todos los vectores de decisión x que satisfacen las $m + p$ restricciones se le conoce como Conjunto de Soluciones Factibles y se denota por X_f . Su imagen se conoce como Región Factible del Espacio Objetivo y se denota por Y_f [3]. Cada vector de decisión es de la forma $x = (x_1, x_2, \dots, x_n)$ y su correspondiente vector objetivo es de la forma $y = (f_1(x), f_2(x), \dots, f_n(x))$. Por lo tanto, el problema consiste en encontrar la x que proporcione el mejor valor para $f(x)$. Sin embargo, en optimización multiobjetivo la mejor solución generalmente no es única, sino que existe un conjunto de mejores soluciones conocidas como soluciones no dominadas. Por lo tanto, es necesario definir el concepto de óptimo en el contexto de la optimización de múltiples objetivos.

Una solución $x \in X_f$ es Pareto óptima si y solo si es no dominada, es decir, si no existe ninguna otra solución $x' \in X_f$ que la domine. Se dice que una solución $x \in X_f$ domina a otra solución $x' \in X_f$ si se cumple que [19]

$$\forall i \in \{1, 2, \dots, k\}, f_i(x) \leq f_i(x') \wedge \exists j \in \{1, 2, \dots, k\} \mid f_j(x) < f_j(x') \quad (3.2)$$

Para denotar que x domina a x' , se escribe $x \succ x'$.

El conjunto de todas las soluciones Pareto Óptimas es denominado Conjunto Pareto Óptimo y matemáticamente puede ser definido como [1]

$$P^* = \{x \in X_f \mid \neg \exists x' \in X_f \text{ para el cual } x' \succ x\} \quad (3.3)$$

La imagen del conjunto Pareto Óptimo se conoce como frente Pareto Óptimo, PF^* .

4. OPTIMIZACIÓN MULTI OBJETIVO EN REDES MULTICAST

Sea $G = (N, E)$ el grafo que modela una red, donde

- N es el conjunto de nodos. El número de nodos de la red es $|N|$.

- E es el conjunto de enlaces.

Sea $s \in N$ el nodo fuente (ingreso), T el conjunto de nodos egreso y $t \in T$ un nodo egreso. El enlace del nodo i al nodo j se representa como (i, j) , donde $(i, j) \in E$. La variable binaria X_{ij}^t indica si el enlace (i, j) es o no usado por el árbol multicast que va desde el nodo ingreso s hasta el conjunto de nodos egreso T . El valor 1 indica que el enlace es usado, y el valor 0 indica lo contrario. Finalmente, el retardo de propagación del enlace (i, j) es denotado como v_{ij} .

El objetivo de nuestro problema es encontrar un árbol multicast que minimice el retardo en la transmisión (*delay*) y el número de saltos (*hop count*). Esto quiere decir que el problema consiste en la minimización de 2 funciones objetivo ($k = 2$), las cuales serán sujetas a las 3 restricciones ($m = 3$) de conservación de flujo. En este caso, cada vector de decisión es de la forma $x = (x_1, x_2, \dots, x_{|T|}) \in X$, donde cada x_t representa la ruta hacia el nodo egreso t . El problema de minimizar el retardo y el número de saltos de un nodo ingreso s a los nodos egresos que pertenecen al conjunto T , se formula matemáticamente de la siguiente manera:

$$\text{Minimizar } w_1 \sum_{t \in T} \sum_{(i,j) \in E} v_{ij} X_{ij}^t + w_2 \sum_{t \in T} \sum_{(i,j) \in E} X_{ij}^t \quad (4.1)$$

$$\text{Sujeto a } \sum_{(i,j) \in E} X_{ij}^t - \sum_{(j,i) \in E} X_{ji}^t = 1, \quad t \in T, i = s \quad (4.2)$$

$$\sum_{(i,j) \in E} X_{ij}^t - \sum_{(j,i) \in E} X_{ji}^t = -1, \quad i, t \in T \quad (4.3)$$

$$\sum_{(i,j) \in E} X_{ij}^t - \sum_{(j,i) \in E} X_{ji}^t = 0, \quad t \in T, i \neq s, i \notin T \quad (4.4)$$

$$\text{donde } X_{ij}^t \in \mathbb{Z}, [0,1] \quad (4.5)$$

En 4.1, el término $\sum_{t \in T} \sum_{(i,j) \in E} v_{ij} X_{ij}^t$ representa el retardo total de propagación sobre todos los enlaces. Sin embargo, esto puede dar como resultado árboles multicast con retardos muy pequeños pero con rutas hacia los destinos demasiado largas. Por este motivo, se incluye en el modelo una función de minimización del número de saltos, representada en 4.1 por el término $\sum_{t \in T} \sum_{(i,j) \in E} X_{ij}^t$.

La restricción 4.2 asegura que el flujo total que sale del nodo ingreso hacia el conjunto de nodos egresos $t \in T$ sea uno. La restricción 4.3, por su parte, asegura que el flujo total que emerge de un nodo egreso $t \in T$ debe ser 1. La restricción 4.4 asegura que para cualquier nodo intermedio la cantidad total de flujo entrante sea igual a la cantidad total de flujo saliente del nodo. Estas restricciones, que se conocen como restricciones de conservación de flujo, aseguran que las rutas hacia cada uno de los nodos egresos se conformen correctamente.

La expresión 4.5 indica que el valor de X_{ij}^t debe ser 0 o 1, dado que no se está haciendo división de un flujo de tráfico en varios subflujos.

5. APLICACIÓN DE LOS ALGORITMOS EVOLUTIVOS EN EL PROBLEMA MULTIOBJETIVO

En esta sección se proponen dos algoritmos evolutivos (NSGA-II, SPEA-II) utilizados en la resolución del problema multiobjetivo presentado anteriormente. Ambos algoritmos reciben como parámetros la topología de la red $G(N, E)$, el nodo ingreso s y el conjunto de nodos egreso T . Antes de presentar los algoritmos describimos la forma como se representan los individuos, la inicialización de la población y los parámetros generales que consideraremos en estos algoritmos evolutivos.

Representación de un individuo. Como queremos encontrar los árboles con valores mínimos de retardo y saltos, es necesario definir cómo vamos a representar cada individuo (árbol multicast) de la población. En este caso se ha considerado un vector de enteros en el que se enumeran cada uno de los nodos que forman un camino desde el nodo origen hasta uno de los nodos destino. La aparición de un -1 en el vector indica que se ha llegado al final de un camino.

La figura 1 representa la red NSF y la solución correspondiente a un flujo que tiene como origen $s = N0$ y como conjunto de nodos destino $T = \{7, 11\}$. El vector de enteros de la figura 2 representa el árbol multicast.

Inicialización de la población. Para inicializar la población es necesario realizar una búsqueda previa de todas las rutas desde el origen hasta cada uno de los nodos destino que pertenecen al conjunto T . Esta búsqueda se realiza utilizando una técnica de búsqueda exhaustiva, conocida como *Backtracking*, que consiste en recorrer las aristas de cada uno de los nodos de la topología hasta encontrar todos los caminos a través del nodo. Luego, se construye el

individuo seleccionando aleatoriamente una ruta para cada uno de los nodos destino. Si el individuo construido ya forma parte de la población inicial, se descarta y se construye uno nuevo. El proceso se repite hasta que la población sea de tamaño N .

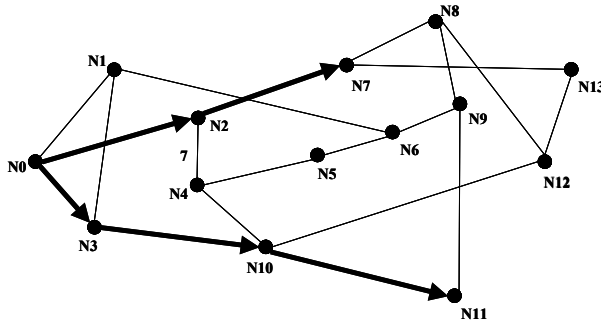


Figura 1. Red NSF

| | | | | | | | | |
|---|---|---|----|---|---|----|----|----|
| 0 | 2 | 7 | -1 | 0 | 3 | 10 | 11 | -1 |
|---|---|---|----|---|---|----|----|----|

Figura 2. Representación de una solución x

Parámetros generales de los algoritmos evolutivos. En general, los parámetros utilizados por NSGA-II y SPEA-II son los siguientes: Tamaño de la población, Tamaño de la población élite, Número máximo de generaciones (g_{max}), Número máximo de generaciones de convergencia (g_{conv}), Probabilidad de Cruce (P_{cruce}) y Probabilidad de Mutación ($P_{mutación}$)

Existen diferentes técnicas para resolver los problemas multiobjetivo; en concreto, en este artículo vamos a utilizar NSGA-II y SPEA-II.

NSGA-II. Propuesto por Deb, Agrawal, Pratap y Meyarivan en el 2000 [5], con el fin de incorporar elitismo y reducir la complejidad del procedimiento de ordenamiento rápido por no dominancia de su antecesor. Realiza una clasificación de la población por frentes. Los individuos que pertenecen al primer frente son los no dominados; los que pertenecen al segundo frente son los no dominados en ausencia de los del frente anterior, y así sucesivamente. A cada individuo se le asigna un rango equivalente a su nivel de no dominancia. Los mejores individuos son aquellos que tienen rangos menores. También incorpora el cálculo de una distancia de *crowding*, como el operador utilizado para mantener la diversidad de la población, con el fin de evitar el uso del σ_{share} en la compartición de aptitud (*fitness sharing*) de su antecesor. La selección es

realizada mediante torneo binario, utilizando como criterio de comparación el operador \succ_n . Según este criterio, el torneo lo gana el individuo con menor rango. Si el rango es el mismo, el torneo lo gana aquel individuo que tenga menor distancia *crowding*.

El pseudocódigo 1 muestra el NSGA-II propuesto en este artículo en su forma general.

Tabla 3
Pseudocódigo del NSGA-II

```

Algoritmo NSGA-II ( $G(N,E), s, T$ )
Inicio
  Para (cada  $t \in T$ ) haga
    Mientras Que (exista un ruta del nodo  $s$  a nodo  $t$ ) haga
      Conjunto_Rutas  $\leftarrow$  Ruta desde  $s$  hasta  $t$ ;
    Fin Mientras Que
  Fin Para
  Generar aleatoriamente  $P_0$ 
  Hacer ordenamiento rápido no dominado de  $P_0$ 
  Aplicar los operadores de selección, cruce y mutación para generar una población hija  $Q_0$ 
  Hacer  $t \leftarrow 1$ 
  Hacer  $R_t \leftarrow \emptyset$ 
  Mientras Que  $t < g_{max}$  y  $Cont\_Convergencia < g_{conv}$ 
    Hacer  $R_t \leftarrow P_t \cup Q_t$ 
    Calcular el número de saltos y el retardo de los miembros  $R_t$ 
     $F \leftarrow OrdenamientoRápidoNoDominado (R)$ 
    Mientras Que ( $|P_{t+1}| < N$ ) hacer
       $CalcularDistanciaCrowding (F_i)$ 
       $P_{t+1} \leftarrow P_{t+1} \cup F_i$ 
    Fin Mientras Que
    Ordenar  $P_{t+1}$  en forma descendente, utilizando el operador  $\succ_n$ 
    Escoger los  $N$  primeros elementos de  $P_{t+1}$ .
    Generar  $Q_{t+1}$  aplicando los operadores de selección cruce y mutación sobre  $P_{t+1}$ .
    Hacer  $t \leftarrow t + 1$ 
  Fin Mientras Que
Fin
    
```

En el algoritmo propuesto, los procedimientos de ordenamiento rápido por no dominancia y cálculo de distancia *crowding* y el operador \succ_n se implementaron de acuerdo a como se describe en [5].

SPEA-II. Desarrollado por Zitzler, Laumanns y Thiele en el 2000 [23] con el fin de superar debilidades detectadas en el esquema de asignación de adaptación

del SPEA [24]. En este algoritmo, la función de asignación de aptitud se mejora teniendo en cuenta para cada individuo el número de individuos a los que domina y el número de individuos por los que es dominado [7]. Este esquema también añade una estimación de densidad poblacional. El tamaño $-N_{E_{max}}$ - de la población externa P_E (utilizada para elitismo) es fijo, a diferencia del SPEA, en el cual el tamaño de P_E es variable pero acotado. P_E está conformada sólo por individuos no dominados siempre y cuando el número de éstos sea mayor o igual que $N_{E_{max}}$. En el caso en que el número de individuos no dominados sea menor que $N_{E_{max}}$ se incluyen individuos dominados dentro de P_E hasta que el tamaño de P_E sea igual a $N_{E_{max}}$. La técnica de agrupamiento (*clustering*), encargada de mantener la diversidad de la población en SPEA, es sustituida por un método de truncamiento, el cual evita eliminar las soluciones extremas del conjunto de soluciones no dominadas. La selección se realiza mediante torneo binario, tomando como criterio de comparación el *fitness* de cada uno de los individuos. SPEA-II asume minimización de *fitness*, por lo tanto gana el torneo aquel individuo que tenga un menor valor de *fitness*.

Los procedimientos de asignación de *fitness*, selección ambiental y truncamiento se implementaron de la forma descrita en [23].

El pseudocódigo 2 muestra el SPEA-II en su forma general.

Tabla 4
Pseudocódigo del SPEA-II

```

Algoritmo SPEA-II ( $G(N,E), s, T$ )
Inicio
  Para (cada  $t \in T$ ) haga
    Mientras Que (exista un ruta del nodo  $s$  a nodo  $t$ ) haga
      Rutas  $\leftarrow$  Ruta desde  $s$  hasta  $t$ 
    Fin Mientras Que
  Fin Para
  Generar aleatoriamente  $P_0$ 
  Hacer  $P_E \leftarrow \emptyset$ 
  Hacer  $t \leftarrow 0$ 
  Mientras Que  $t < g_{max}$  y Cont_Convergencia  $< g_{conv}$ 
    Calcular el número de saltos y el retardo de los miembros de  $P^t$  y  $P_E^t$ 
    Calcular el fitness de cada uno de los individuos en  $P^t$  y  $P_E^t$ 
    Realizar un proceso de selección ambiental para conformar la nueva población externa  $P_E^{t+1}$ 
    Aplicar el operador de selección por torneo binario con reemplazo sobre  $P_E^{t+1}$ .
    Aplicar los operadores de cruce y mutación sobre la población seleccionada.
    Asignar la nueva generación a  $P^{t+1}$ .
    Hacer  $t \leftarrow t + 1$ 
  Fin Mientras Que
Fin
    
```

6. EXPERIMENTACIÓN Y RESULTADOS

6.1 Diseño del experimento

Como ya hemos mencionado previamente, los algoritmos evolutivos seleccionados para resolver el problema propuesto son NSGA-II y SPEA-II. Para obtener unos resultados fiables se realizaron 30 ejecuciones sucesivas de cada uno de ellos en tres topologías de red diferentes:

- NSF. Topología de red de la National Science Foundation. Está conformada por 14 nodos y 40 enlaces
- T25. Topología generada aleatoriamente. Consta de 25 nodos y 150 enlaces
- T100. Topología generada aleatoriamente. Consta de 100 nodos y 1000 enlaces.

El número de nodos destino variaba entre 2 y 10. Para cada conjunto de nodos destino se calculó el tiempo promedio de ejecución del algoritmo, así como los valores máximo, mínimo y promedio de del retardo y el número de saltos.

Para ambos algoritmos, el tamaño de la población es de 50, el tamaño de la población élite es 25, el número máximo de generaciones (g_{max}) es 50, el número de generaciones de convergencia (g_{conv}) es 5, P_{cruce} es 0.8 y $P_{mutación}$ es 0.1. En ambos casos se implementó el mismo criterio de convergencia: el algoritmo se detiene si se superan más del 10% del máximo de generaciones sin que aparezcan nuevos individuos en la población élite.

Todas las ejecuciones se realizaron en un computador personal con un procesador AMD Athlon – 1Ghz, con 256 Mb de memoria RAM.

Para poder determinar la exactitud de los algoritmos evolutivos (AEs) propuestos, el modelo multiobjetivo presentado en la sección 4 se resolvió utilizando el solver SNOPT de GAMS [12].

6.2 Análisis de resultados

La gráfica de las figuras 3a y 3b muestran los valores mínimos de la función número de saltos dados por el modelo analítico, así como los obtenidos mediante los algoritmos evolutivos para diferentes conjuntos de nodos destino, para NSF y T25. Es claro que para NSF estos valores son más exactos que para

T25, especialmente para conjuntos T de cardinalidad mayor que 5; esto se debe a que a medida que el tamaño de la topología y el número de nodos destino aumentan, el espacio de búsqueda se hace más grande y, por lo tanto, la aproximación al conjunto de soluciones reales disminuye. Sin embargo, los valores obtenidos mediante los algoritmos evolutivos para T25 son muy cercanos a los valores analíticos, lo cual indica que los AEs resultan bastante exactos a la hora de encontrar el mínimo número de saltos en NSF y T25.

Viendo que en las topologías pequeñas (NSF y T25) los resultados obtenidos mediante los algoritmos evolutivos se aproximan muy bien a los valores dados por el modelo analítico, pasamos a experimentar los algoritmos en una topología más grande (T100), con la certeza de que si bien es muy probable que los valores obtenidos no sean exactos, sí estarán muy cercanos a los reales. La figura 3c muestra los valores obtenidos mediante NSGA-II y SPEA-II para la función número de saltos. Este resultado nos confirma nuevamente que ambos algoritmos se comportan de una manera similar frente al modelo propuesto, independientemente de la topología sobre la cual se esté trabajando.

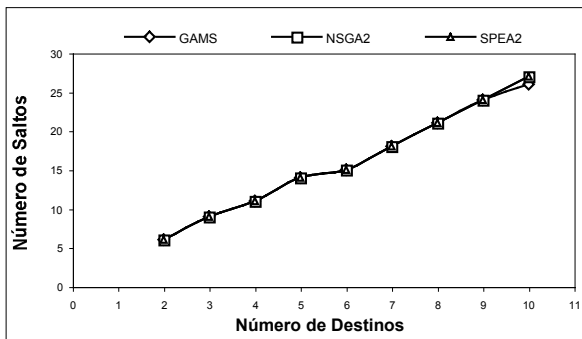


Figura 3a. Mínimo N° de saltos NSF

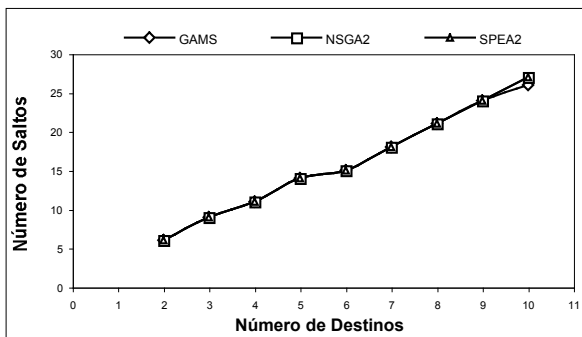


Figura 3b. Mínimo N° de saltos T25

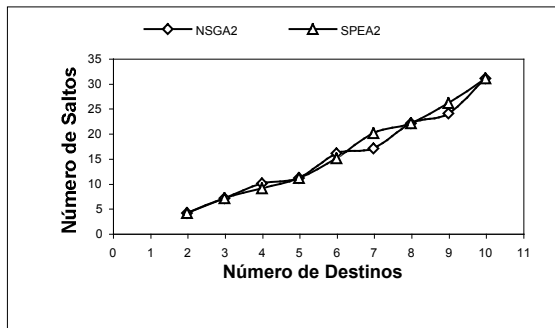


Figura 3c. Mínimo N° de saltos T100

Las figuras 4a y 4b, al igual que en el caso del mínimo número de saltos, muestra que los valores mínimos del retardo en la transmisión se empiezan a distorsionar a partir de 9 y 6 nodos destino para NSF y T25 respectivamente, lo cual nos confirma que la aproximación disminuye un poco cuando el tamaño de la topología y el número de nodos destino aumenta, sin dejar de ser muy buena debido a la cercanía de los valores experimentales con los valores obtenidos analíticamente. Por lo tanto, respecto al mínimo retardo en la transmisión también podemos decir que la aproximación de los algoritmos evolutivos a los valores reales es bastante exacta.

En el caso de T100, la figura 4c muestra que los resultados obtenidos para la función retardo en la transmisión por NSGA-II y SPEA-II son muy similares, por lo tanto, en este caso tampoco podemos afirmar que un algoritmo se comporte mejor que otro. La exactitud de los resultados obtenidos para NSF y T25 nos aseguran que los resultados obtenidos en T100 para el retardo son, al igual que para la función número de saltos, una buena aproximación a los valores obtenidos mediante la solución analítica.

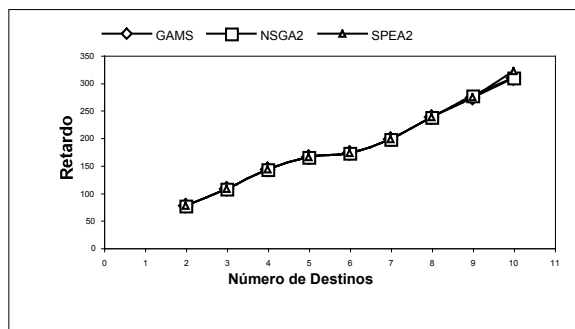


Figura 4a. Retardo Mínimo NSF

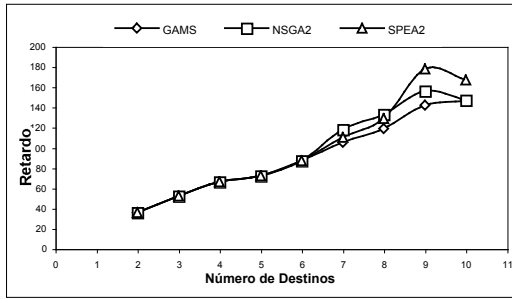


Figura 4b. Retardo Mínimo T25

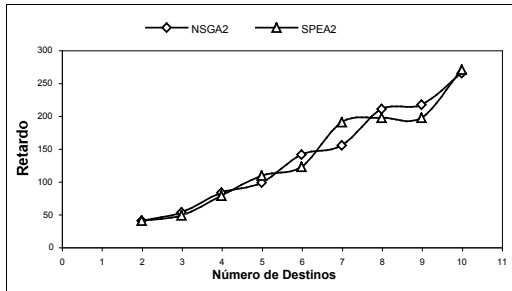


Figura 4c. Retardo Mínimo T100

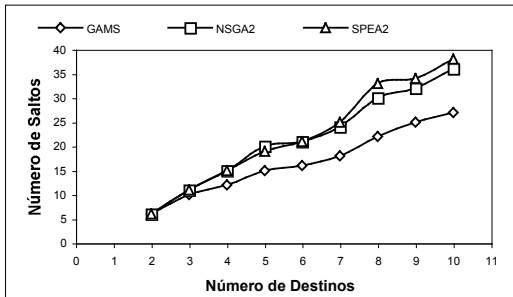


Figura 5a. Máximo número de saltos NSF

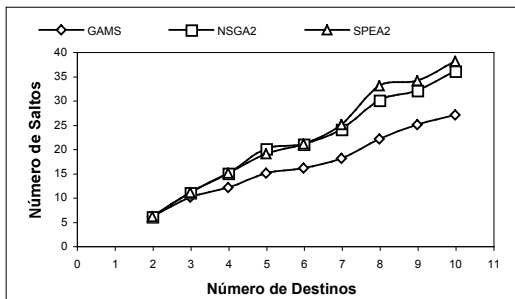


Figura 5b. Máximo número de saltos T25

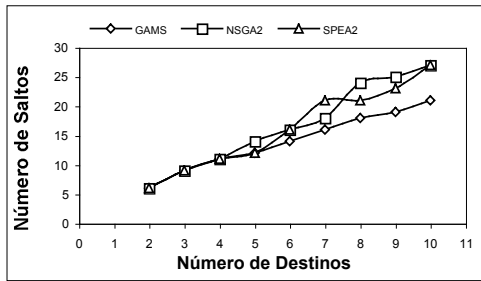


Figura 5c. Máximo número saltos T100

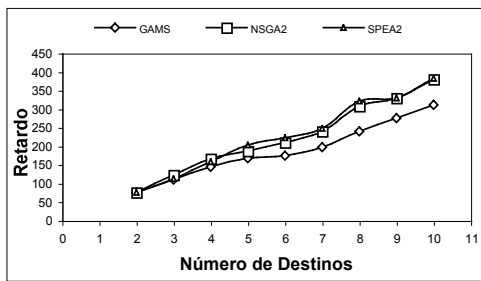


Figura 6a. Retardo Máximo NSF

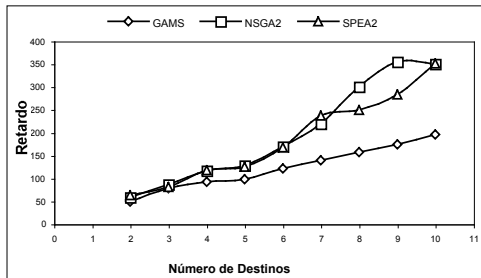


Figura 6b. Retardo Máximo T25

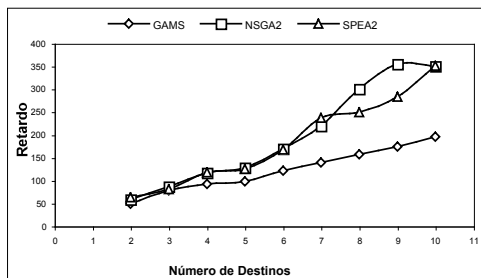


Figura 6c. Retardo Máximo T100

Como se aprecia en las figuras 5a, 5b, 6a y 6b, los valores máximos obtenidos en NSF y T25 para cada una de las funciones mediante los algoritmos evolutivos, se empiezan a distorsionar más rápido que los valores mínimos. Sin embargo, estos valores siguen siendo bastante cercanos a los analíticos, y por lo tanto, en este caso también podemos decir que la capacidad de los algoritmos para generar valores máximos de ambas funciones es muy buena.

En 5c y 6c se observa que nuevamente NSGA-II y SPEA-II se comportaron de manera muy similar en la generación de valores máximos de ambas funciones objetivo.

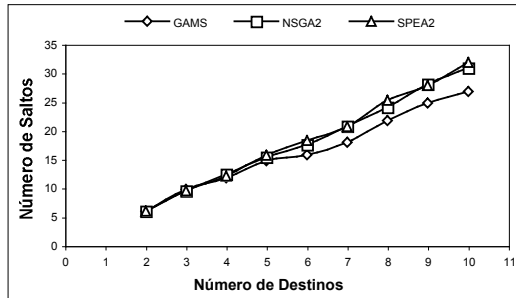


Figura 7a. Número de saltos promedio NSF

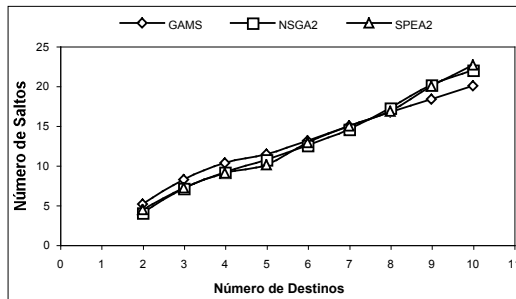


Figura 7b. Número de saltos promedio T25

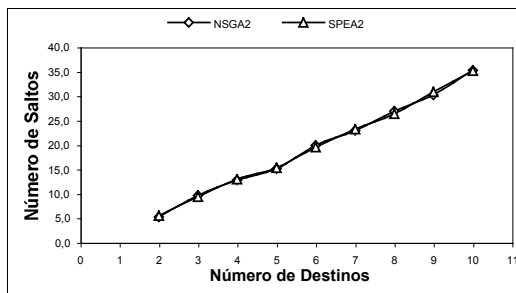


Figura 7c. Número de saltos promedio T100

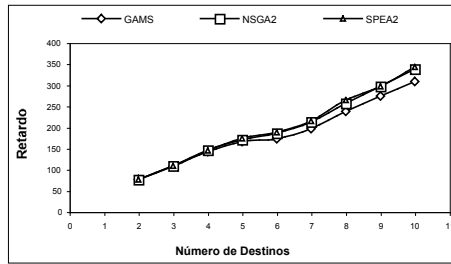


Figura 8a. Retardo promedio NSF

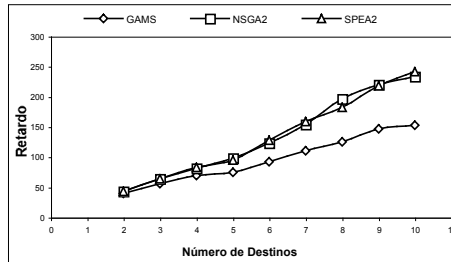


Figura 8b. Retardo promedio T25

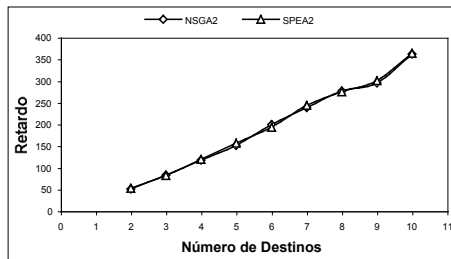


Figura 8c. Retardo promedio T100

En las figuras 7a, 7b, 8a y 8b se aprecian los valores promedio analíticos para el número de saltos y el retardo en la transmisión, así como los valores promedio obtenidos con los algoritmos evolutivos. Resulta evidente que los valores experimentales son bastante cercanos a los analíticos, lo cual confirma una vez más que los algoritmos experimentados están en capacidad de aproximarse muy bien a los valores de las funciones que están optimizando, obtenidos analíticamente. Por su parte, las figuras 7c y 8c muestran que los valores promedio de las funciones objetivo obtenidos experimentalmente con NSGA-II y SPEA-II, al igual que los valores máximos y mínimos de las mismas, son muy cercanos entre sí, lo cual refleja una vez más que ambos algoritmos son igualmente eficaces en la generación de soluciones óptimas.

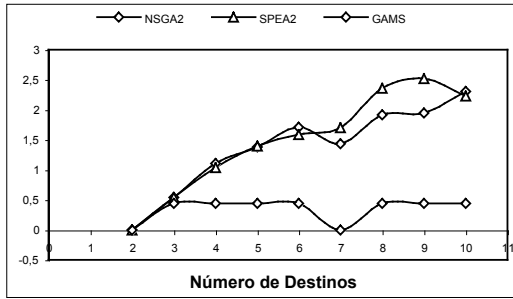


Figura 9a. Desv. estándar número de saltos NSF

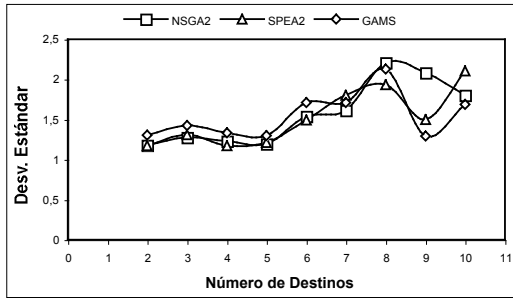


Figura 9b. Desv. estándar número de saltos T25

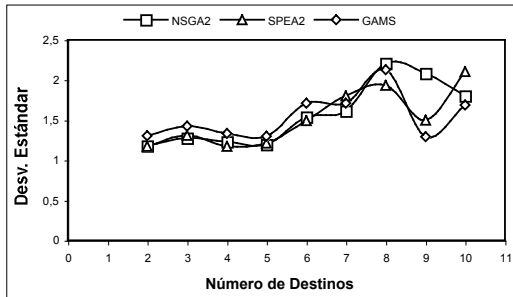


Figura 9c. Desv. estándar número de saltos T100

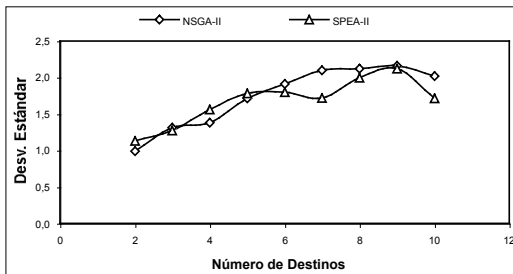


Figura 10a. Desv. estándar retardo NSF

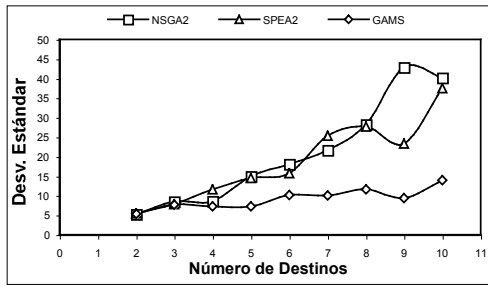


Figura 10b. Desv. estándar retardo T25

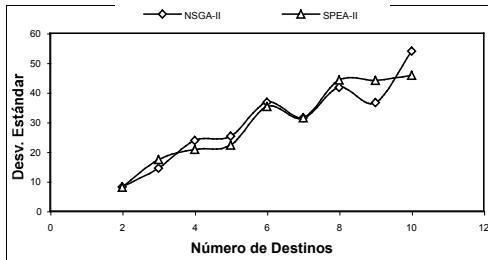


Figura 10c. Desv. estándar retardo T100

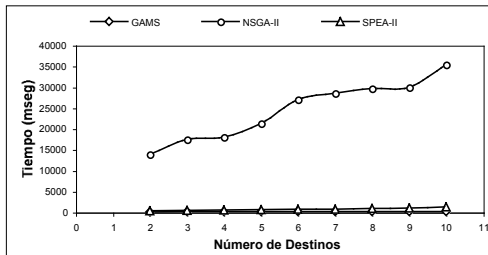


Figura 11a. Tiempo promedio NSF

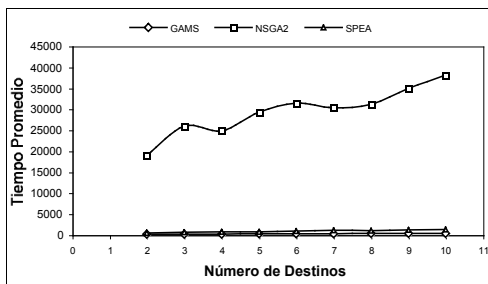


Figura 11b. Tiempo promedio T25

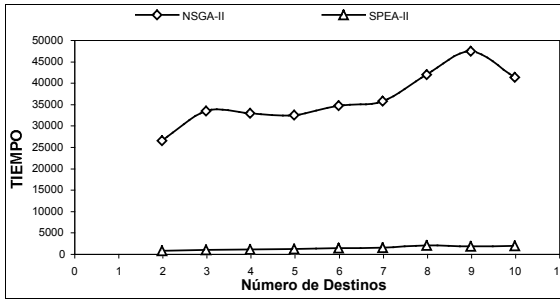


Figura 11c. Tiempo promedio T100

Las gráficas de las figuras 11a, 11b y 11c muestran el comportamiento de los tiempos de respuesta promedio de cada uno de los algoritmos frente a las tres topologías diferentes. Como era de esperarse, estos tiempos se incrementan a medida que los algoritmos se aplican a topologías con mayor número de nodos. Por lo tanto, lo interesante de las gráficas es la gran diferencia presentada por los tiempos empleados por SPEA-II frente a los de NSGA-II. En las tres figuras se aprecia claramente que mientras para GAMS y SPEA-II emplean tiempos promedio muy inferiores a los 5.000 milisegundos, los tiempos empleados por NSGA-II están por encima de los 10.000 milisegundos en NSF, 15.000 milisegundos en T25 y 25.000 milisegundos en T100. Lo anterior nos permite afirmar que SPEA-II es mucho más eficiente en la solución del problema que su oponente independientemente del tamaño de la topología sobre la que se está trabajando.

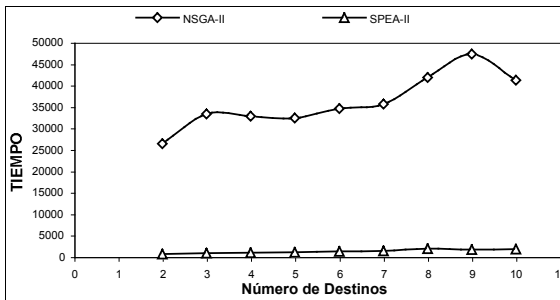


Figura 12a. T25 - 3 destinos

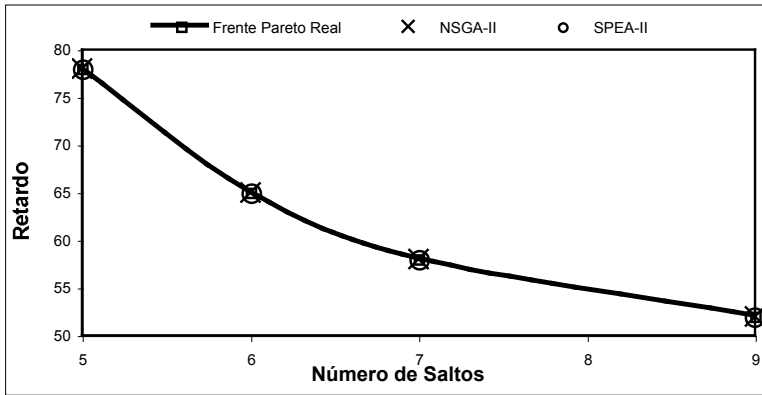


Figura 12b. T25 - 5 destinos

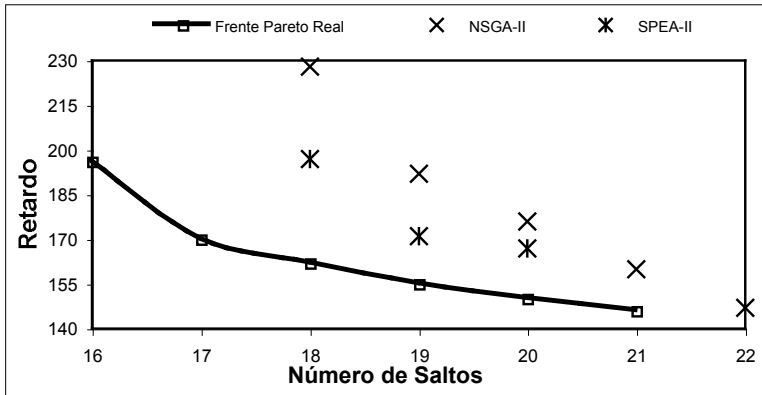


Figura 12c. T25 - 7 destinos

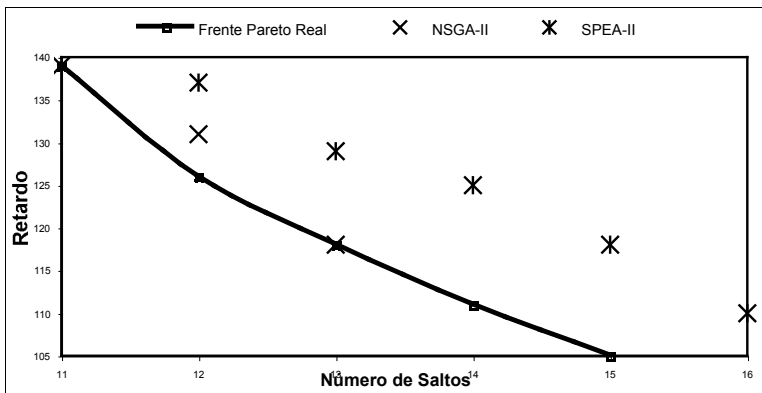


Figura 12d. T25 - 10 destinos

Las figuras 12 a, 12b, 12c y 12d muestran 4 frentes de Pareto óptimo para conjuntos de 3, 5, 7 y 10 nodos destino en T25. El conjunto de soluciones Pareto óptimas para NSGA-II y SPEA-II se conformó mediante la unión de las mejores soluciones obtenidas durante las 30 ejecuciones de los algoritmos. Como se aprecia en 12a y 12b, la aproximación de los algoritmos al frente de Pareto óptimo analítico para conjuntos de 3 y 5 nodos destino es exacta; en el caso de las figuras 12c y 12d se puede observar que las soluciones si bien no son exactas, sí están bastante próximas al frente Pareto analítico.

CONCLUSIONES

En este artículo se ha presentado la técnica de optimización de objetivos múltiples conocida como algoritmos evolutivos multiobjetivo (MOEAs), mediante la aplicación de dos de los algoritmos evolutivos más conocidos, NSGA-II y SPEA-II.

El problema multiobjetivo escogido como referencia para determinar la eficacia y la eficiencia de los algoritmos, consiste en encontrar un árbol multicast que minimice el número de saltos y el retardo en la transmisión, para 3 topologías de red de tamaños diferentes. Para comparar resultados y determinar la exactitud de los algoritmos evolutivos, se resolvió el modelo multiobjetivo utilizando el solver SNOPT de GAMS sobre dos de las topologías de red (NSF, T25).

De acuerdo a los resultados obtenidos para las topologías NSF y T25, se pudo comprobar que ambos algoritmos tienen capacidades similares para encontrar soluciones cercanas al conjunto de soluciones Pareto óptimas obtenidas analíticamente, dado que dado que si bien es cierto que para algunos conjuntos de destinos el desempeño de uno es mejor que el del otro, la diferencia entre las soluciones obtenidas por cada uno de ellos y su convergencia hacia el frente de Pareto no es tan significativa como para inclinar la balanza a favor de uno u otro algoritmo.

La topología T100 no pudo ser comparada con una solución analítica debido a la complejidad que implica el modelado de la misma. Sin embargo, los resultados obtenidos para NSF y T25 nos permiten decir que si bien éstos no serán exactos, sí estarán bastante cercanos a los analíticos. Los resultados obtenidos para T100 muestran un comportamiento similar de las funciones retardo y número de saltos en ambos algoritmos.

En este trabajo se comprobó que SPEA-II resultó ser mucho más eficiente en términos del tiempo de convergencia hacia las soluciones no dominadas, independientemente del tamaño de la topología y del número de nodos que se estuviera optimizando. Mediante una prueba Z de diferencia de medias muestrales se confirmó, con un nivel de confianza del 95%, que, independientemente del tamaño de la topología y del número de nodos destino, SPEA-II resultó ser mucho más eficiente que NSGA-II.

Mediante los experimentos realizados también se comprobó que la convergencia de las soluciones obtenidas por los algoritmos evolutivos hacia la frontera de Pareto analítica disminuye a medida que el conjunto de nodos destinos aumenta. Esto se debe a que el número de posibles combinaciones de rutas se incrementa a medida que aumenta el número de destinos, y por lo tanto el espacio de búsqueda se hace más grande. Además, estadísticamente se comprobó, mediante una prueba Z de diferencia de medias muestrales, con un nivel de confianza del 95%, que en promedio la convergencia de las soluciones obtenidas es la misma para ambos algoritmos, independientemente del tamaño de la topología.

El problema de decidir cuál de los dos algoritmos resulta mejor para resolver el problema multiobjetivo propuesto es en sí mismo un problema de optimización de objetivos múltiples, dado que mejorar un determinado aspecto como la eficacia de los algoritmos, es decir, la exactitud de las soluciones halladas por éstos, puede llevar a desmejorar otro aspecto también muy importante como es la eficiencia. Sin embargo, en este estudio se ha demostrado que SPEA-II resultó ser igual de eficaz que NSGA-II pero mucho más eficiente que este último, por lo tanto SPEA-II resulta mejor en la solución del problema propuesto, dado que en términos de dominancia de Pareto iguala a NSGA-II en un objetivo (exactitud de las soluciones) y lo mejora en el otro (eficiencia); en otras palabras, SPEA-II domina a NSGA-II.

Referencias

- [1] ALMEIDA, C.; AMARILLA, N.; BARÁN, B. (marzo, 2003). Optimización Multiobjetivo en la planificación de Centrales Telefónicas. Universidad Nacional de Asunción, Campus Universitario de San Lorenzo Paraguay.
- [2] BANERJEE, N; DAS, S.K. (2001). Fast Determination of QoS Multicast Routes in Wireless Networks using Genetic Algorithm. Communications. ICC 2001.
- [3] BARÁN, B.; DUARTE, S. (2000). Multiobjective Network Design Optimization using Parallel Evolutionary Algorithms. Centro Nacional de Computación, Universidad Nacional de Asunción.

- [4] CUI, X ; LIN, C ; WEI, Y. A Multiobjective Model for Qos Multicast Routing Based on Genetic Algorithm. ICCNMC'03
- [5] DEB K.; AGRAWAL, S.; PRATAP, A.; MEYARIVAN, T. (2000). A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In KanGAL Report N° 200001. Indian Institute of Technology, Kanpur, India.
- [6] DEB, K. (2001). Multiobjective Optimization Using Genetic Algorithms. John Wiley & Sons, Chichester, UK.
- [7] DIAS, A.; DE VASCONCELOS J. (March, 2002). "Multiobjective Genetic Algorithms Applied to Solve Optimization Problems". IEE Transactions on Magnetics, Vol. 38, N° 2.
- [8] DONOSO, Y; FABREGAT, R; FABREGA, L. Multi-Objective scheme over multi-tree routing in multicast MPLS networks. ACM/IFIP LANC'03.
- [9] DONOSO, Y; FABREGAT, R; MARZO, J.L. Multi-Objective Optimization Model and Heuristic Algorithm for Multipath Routing of Dynamic Multicast Groups IEEE ISCC'04
- [10] DONOSO, Y; FABREGAT, R; MARZO, J.L. Multi-objective optimization algorithm for multicast routing with traffic engineering. IEEE ICN'04.
- [11] FABREGAT, R; DONOSO Y; MARZO, J.L; ARIZA, A. A Multi-Objective Multipath Routing Algorithm for Multicast Flows. SPECTS'04".
- [12] GALVÁN, B.; GREINER, D.; WINTER, G. (2003). "Una comparativa de Algoritmos Evolutivos Multicriterio en Diseño de Sistemas de Seguridad". Segundo Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados. MAEB03, Gijón.
- [13] GAMS. Solver for large mathematical programming problems. <http://www.gams.com>.
- [14] INAGAKI, J; HASEYAMA, M; KITAJIMA, H. (1999). A Genetic Algorithm for Determining Multiple Routes and its Applications. Circuits and Systems. ISCAS '99.
- [15] LEUNG, Y ; LI, G ; XU, Z.B. (November, 1998). A Genetic Algorithm for the Multiple Destination Routing Problems. IEEE Transactions on Evolutionary Computation, Vol. 2, N° 4.
- [16] LI, Y; BOUCHEBABA, Y. (1999). A new genetic algorithm for the optimal communication spanning tree problem. In C. Fonlupt, J.-K. Hao, E. Lutten, E. Ronald, and M. Schoenauer, editors, Proceedings of Artificial Evolution: Fifth European Conference, volume 1829 of LNCS, p. 162-173. Springer.
- [17] MILLER, K. (1999). "Multicast Networking and Applications". Ed. Addison Wesley Longman, Inc.
- [18] ROY, A; DAS, S.K. (May, 2004). QM2RP: A QoS-based Mobile Multicast Routing Protocol using Multi-Objective Genetic Algorithm. The Journal of Mobile Communication, Computation and Information, Wireless Networks, Kluwer.
- [19] SALAZAR, D. (November, 2003). "Evaluación de métodos evolutivos de optimización multiobjetivo de 2ª generación". Universidad Central de Venezuela.
- [20] SRINIVAS, N; DEB, K. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. IEEE Journal of Evolutionary Computation, Vol. 2, N° 3, p. 221-248.
- [21] SUN, Q. (1999). A genetic algorithm for delay-constrained minimum-cost multicasting. Technical Report, IBR, TU Braunschweig, Butenweg, 74/75, 38106, Braunschweig, Germany.

- [22] XIANG, F; JUNZHOU, L; JIEYI, W; GUANQUN, G. (May, 1999). QoS routing based on genetic algorithm. *Computer Communications* 22. pp. 1293-1399. Elsevier.
- [23] ZITZLER, E.; LAUMANN, M.; THIELE, L. (2000). "SPEA2: Improving the Strength Pareto Evolutionary Algorithm". Computer Engineering and Networks Laboratory (TIK). Department of Electrical Engineering Swiss Federal Institute of Technology (ETH) Zurich.
- [24] ZITZLER, E. & THIELE, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation* 3(4), 257-271.