

Minimización del retraso extremo a extremo en redes Unicast utilizando técnicas metaheurísticas

Minimizing the end-to-end delay in Unicast networks using metaheuristic techniques

V. L. Hernández, J. M. López, H. E. Hernández

Recibido Junio 20 de 2013 – Aceptado Noviembre 15 de 2013

Resumen - En el presente trabajo se adopta una nueva metaheurística (EBOLA) para resolver el problema de Enrutamiento Unicast QoS, como una nueva alternativa para resolver este tipo de problemas. Los resultados de ejecución muestran que el algoritmo propuesto es tan eficiente como Algoritmo Genético para enrutamiento QoS.

Palabras Clave - QoS, enrutamiento Unicast, algoritmo genético.

Abstract - This paper adopts a new metaheuristic (Ebola) to solve the QoS Unicast Routing Problem. We performed a hypothesis test to compare the delays obtained by the proposed algorithm compared to genetic algorithm in solving instances. The implementation results show that the proposed algorithm is an efficient algorithm for QoS routing.

Key Words - QoS, Unicast routing, genetic algorithm.

¹ Producto derivado del proyecto de Investigación “Enrutamiento y calidad de servicio QoS en redes de computadores usando técnicas metaheurísticas”, apoyado por la Universidad de Córdoba a través de los grupos de investigación GI – TOC y Sócrates.

V. L. Hernández. Ingeniería de Sistemas, Magíster en Teleinformática, docente Departamento de Ingeniería de Sistemas y Telecomunicaciones, Universidad de Córdoba, Colombia, grupo de investigación Sócrates, (correos e.: velssyliliana@gmail.com).

J. M. López. Ingeniería Industrial, Maestría en Ingeniería Industrial, docente, Departamento de Ingeniería Industrial, Universidad de Córdoba, Colombia, grupo de investigación en la aplicación de teoría de restricciones GI – TOC, (correo e.: jotamlopez@gmail.com).

H. E. Hernández. Ingeniería Industrial, Especialización en Gerencia de Producción y Calidad, Maestría en Gestión de Organizaciones, (c) Doctorado en Ingeniería Industrial, docente, Departamento de Ingeniería Industrial, Universidad de Córdoba, Colombia, Grupo de investigación en la aplicación de teoría de restricciones GI – TOC, (correos e.: hhernandez@correo.unicordoba.edu.co).

I. INTRODUCCION

EN una red de datos pueden existir diversas rutas para enviar tráfico de un nodo origen a un nodo destino. Sin embargo, encontrar una ruta óptima que permita ofrecer un mejor servicio de comunicación dependiendo de algún criterio de desempeño, es un problema a resolver. Cuando el criterio de desempeño está relacionado con métricas como retraso, ancho de banda, costo; entre otros, nos referimos entonces específicamente a un problema de enrutamiento Quality of Service QoS [1]. Además, cuanto más grande sea el número de nodos interconectados, mayor será la dificultad para encontrar la mejor solución. La selección de la ruta en enrutamiento es típicamente planteada como un problema de optimización del camino más corto, como por ejemplo, determinar una serie de enlaces de red que conectan una fuente al destino tal que una función objetivo en particular es minimizada. La función objetivo puede ser el número de saltos, costo, retraso, o alguna otra métrica que corresponda a la suma numérica de parámetros de enlace individual a lo largo de la ruta seleccionada. Satisfacer diversos requerimientos de QoS dificulta los cálculos en la medida en que se involucran más restricciones en el problema de optimización [1], [2]. El problema de encontrar la ruta más corta con aún una sola restricción de ruta, como por ejemplo, el mínimo retraso extremo a extremo a través de la red, es considerado un problema NP-Complete [1]. Este tipo de problemas se resuelve mediante el uso metaheurísticas como algoritmos evolutivos, colonia de hormigas, recocido simulado, búsqueda tabú, etc. [3]. Sin embargo, estos algoritmos no garantizan la obtención del óptimo global, pero pueden encontrar buenas soluciones, teniendo en cuenta los tiempos de cálculo. Algoritmo Genético ha sido utilizado ampliamente en una variedad de problemas NP, incluyendo una amplia aplicación en problemas de enrutamiento QoS [3].

Diversos trabajos se han desarrollado para problemas de optimización en enrutamiento unicast utilizando algoritmo genético: [4] proponen un nuevo planteamiento de enrutamiento QoS para aplicaciones multimedia usando Algoritmo Genético, teniendo en cuenta como parámetro de enrutamiento el tiempo de retraso. En [5] se propone un nuevo algoritmo primal-dual para encontrar una ruta con el menor costo utilizando el tiempo de retraso como restricción. En [3] se utiliza un nuevo algoritmo genético híbrido para resolver el problema de enrutamiento unicast en redes en malla inalámbrica, minimizando el costo y utilizando múltiples restricciones. En [6] desarrollan una nueva metaheurística basada en Algoritmo Genético para encontrar rutas factibles satisfaciendo múltiples restricciones en menor tiempo.

En el presente trabajo se adopta una nueva metaheurística (EBOLA) para resolver el problema de Enrutamiento Unicast QoS, como una nueva alternativa para resolver este tipo de problemas. Los resultados de ejecución muestran que el algoritmo propuesto es tan eficiente como Algoritmo Genético para enrutamiento QoS.

II. PROBLEMA UNICAST

El Problema de enrutamiento puede ser dividido en dos clases principales: enrutamiento Unicast y enrutamiento Multicast. El Problema de enrutamiento Unicast consiste en que dado un nodo fuente , un nodo destino , un conjunto de restricciones de QoS por cumplir y un objetivo que optimizar; debe encontrar el mejor camino factible desde el nodo fuente al nodo destino [4], [7], [8], [9]. El problema de enrutamiento Multicast hace referencia a un nodo fuente, un conjunto de nodos destino, un conjunto de restricciones y una función objetivo, para encontrar la mejor ruta factible que satisface las restricciones, desde el nodo fuente a todos los nodos destino. En este caso el fin es encontrar el árbol que conecta el nodo origen con el conjunto de los nodos destino cumpliendo los requisitos de calidad de servicio [10].

Los requerimientos de QoS de una conexión están dados como un conjunto de restricciones que pueden ser de dos tipos: restricciones de enlace, restricciones de ruta [1]. Una restricción de enlace especifica una restricción sobre el uso de los enlaces. Por ejemplo, una restricción de ancho de banda en una conexión de tipo unicast, requiere que los enlaces que conforman la ruta deben tener cierta cantidad de ancho de banda disponible. Una restricción de ruta, especifica los requerimientos de QoS de extremo a extremo sobre una sola ruta.

A. Modelo Matemático [11]

Para resolver un problema de optimización en redes de datos, es necesario definir una variable que describa la ruta a través de la cual el flujo de información será transmitido. A continuación se presenta la variable

$$X_{ij}^f = \begin{cases} 1, & \text{si el enlace } (i,j) \text{ es usado por el flujo } f \\ 0, & \text{si el enlace } (i,j) \text{ no es usado por el flujo } f \end{cases} \quad (1)$$

TABLA I
PARÁMETROS UNICAST

Términos	Definición
$G(N, E)$	Grafos de la topología
$N!$	Conjunto de nodos !
E	Conjunto de enlaces
s	Nodo fuente
t	Nodo destino
(i, j)	Enlace desde el nodo i al nodo j
F	Conjunto de flujos
f	Cualquier flujo unicast
X_{ij}^f	Indica si el enlace (i,j) es usado por el flujo f con destino al nodo t
c_{ij}	Capacidad disponible de cada enlace (i,j)
bw_f	Demanda de tráfico de un flujo f

Esto significa que la variable del vector X_{ij}^f es 1 si el enlace es usado para transmitir el flujo f ; si no puede ser usado, el vector es 0. Los parámetros utilizados para modelar la transmisión de datos en el caso unicast se presentan en la tabla I.

La función de optimización es el retraso de propagación por enlace (*delay*). La unidad asociada a la función de retraso es el tiempo, expresado en milisegundos. Esta función es acumulativa, ya que el objetivo es calcular y obtener el mínimo retraso extremo a extremo. Teniendo en cuenta la función *delay*, cada enlace tiene un peso d_{ij} que denota el retraso de propagación entre el nodo i y el nodo j .

La optimización de la función *delay* consiste en minimizar la suma del producto de los tiempos de retraso o *delay* d_{ij} de cada enlace (i,j) por el valor asociado con la variable X_{ij}^f . La función para minimizar el retraso extremo a extremo puede ser representado en la siguiente expresión:

$$\min \sum_{f \in F} \sum_{(i,j) \in E} d_{ij} \cdot X_{ij}^f \quad (2)$$

Donde:

$$\sum_{(i,j) \in E} d_{i,j} \cdot X_{i,j}^f \quad (3)$$

Representa el valor del retraso extremo a extremo de las posibles rutas para el flujo f .

La ecuación (4) representa todos los flujos que serán transmitidos sobre la red. Por cada flujo f , se obtendrá una ruta con el mínimo retraso extremo a extremo desde el nodo origen s a su nodo destino t .

$$\sum_{f \in F} \quad (4)$$

Para garantizar que las soluciones obtenidas en una transmisión unicast sean rutas válidas desde el origen s al destino t , es necesario tener en cuenta tres restricciones

de conservación de flujo y una de capacidad del enlace. Teniendo en cuenta que la variable X_{ij}^f indica el uso del enlace (i,j) por el flujo f , consideramos esta variable positiva cuando el enlace deja el nodo, y negativa en el caso opuesto. La primera restricción asegura que para cada flujo f solamente una ruta sale desde el nodo origen s . La ecuación que modela esta restricción está dada por la siguiente expresión:

$$\sum_{(i,j) \in E} X_{ij}^f = 1, \quad f \in F, \quad i = s \quad (5)$$

La segunda restricción asegura que para cada flujo solamente una ruta alcanza el nodo destino t . La ecuación que modela esta restricción está dada por la siguiente expresión:

$$\sum_{(j,i) \in E} X_{ji}^f = -1, \quad f \in F, \quad i = t \quad (6)$$

$$\sum_{(i,j) \in E} X_{ij}^f - \sum_{(j,i) \in E} X_{ji}^f = 0, \quad f \in F, \quad i \neq s, \quad i \neq t \quad (7)$$

Por último, se requiere una restricción relacionada con la demanda de tráfico necesaria para cada flujo de datos y la capacidad de los enlaces. Es necesario prevenir enviar una demanda de tráfico más grande que la capacidad del enlace. La restricción se expresa de la siguiente forma:

$$\sum_{f \in F} bw_f \cdot X_{ij}^f \leq c_{ij}, \quad (i,j) \in E \quad (8)$$

Donde:

$$\sum_{f \in F} bw_f \cdot X_{ij}^f \quad (9)$$

Representa toda la demanda por flujo en bits por segundo transmitida en un enlace (i,j) . La expresión no permite que la demanda sea más alta que la capacidad del enlace.

III. ALGORITMO EVOLUTIVO

Los Algoritmos Evolutivos están entre los tipos de algoritmos utilizados para resolver problemas de optimización combinatorial y cuyas soluciones tradicionales son muy complejas en un tiempo computacional. Se originan en la Teoría de la Evolución de Darwin, y utilizan tres componentes fundamentales de la evolución: replicación, variación y selección natural. Replicación es la formación de un nuevo organismo a partir de uno previo. Durante el proceso de replicación ocurren una serie de errores llamados variaciones que producen cambios en los individuos. Además de replicación y variación, la evolución necesita selección natural, la cual ocurre cuando individuos de la misma especie compiten por los recursos de su ambiente y la posibilidad de reproducción. Tal competencia permite a los individuos más aptos sobrevivir y a los menos aptos morir.

Un algoritmo genético se compone de individuos, la población de tales individuos, el más apto de los individuos y los operadores genéticos, como se muestra en la Fig. 1:

1) Individuo

Un individuo es la representación de una solución del problema.

2) Población

El conjunto de individuos usados por el algoritmo para realizar el proceso de evolución, se denomina población.

3) La Función de Aptitud (Fitness)

Asigna a cada individuo un valor real que muestra qué tan buena es la adaptación de un individuo a su ambiente.

4) Operadores Genéticos

Son aplicados a los individuos de la población, modificándolos para obtener una mejor solución. Los operadores más usados en algoritmos evolutivos son: selección, recombinación y mutación [11].



Fig. 1. Ciclo de Algoritmo Evolutivo (adaptado de [2]). A una población de 'n' individuos se aplica una evaluación para determinar los mejores individuos en función de la aptitud o fitness. De estos mejores individuos se aplica una selección para determinar los individuos que serán elegidos como padres, haciendo que se reproduzcan con ayuda de los operadores genéticos. Estos causan una serie de transformaciones de alteración y recombinación en los individuos, para generar los descendientes. Este ciclo se repite hasta conseguir hacer evolucionar la población.

La estructura de un algoritmo evolutivo es la siguiente:

INICIO //Algoritmo Genético

Generar una población inicial.

Calcular la función de evaluación de cada individuo.

MIENTRAS criterio de parada sea falso HACER

INICIO //Producir nueva generación

PARA Tamaño población HACER

INICIO //Ciclo Reproductivo

Seleccionar dos individuos de la anterior generación, para el cruce (probabilidad de selección proporcional a la función de evaluación del individuo).

Cruzar: con cierta probabilidad los dos individuos obteniendo dos descendientes.

Mutar: los dos descendientes con cierta probabilidad. Computar: la función de evaluación de los dos descendientes mutados. Insertar los dos descendientes mutados en la nueva generación.

FIN INICIO
 SI la población ha convergido HACER Terminado.
 FIN SI
 FIN PARA
 FIN INICIO
 FIN MIENTRAS
 FIN INICIO

I. ALGORITMO ÉBOLA

Los virus son organismos compuestos solo de ácidos nucleicos, lo que implica que necesitan de células vivas para reproducirse [12]. Los virus usan la maquinaria de replicación de DNA de la célula para reproducirse, luego la célula libera los nuevos virus que procederán a infectar nuevos tejidos.

Particularmente el virus de Ébola es un patógeno viral que produce fiebre hemorrágica, clasificado en la familia Filoviridae, con un solo género Filovirus, perteneciente al orden Mononegaviral [13]. De la misma manera, Ébola computacional tiene procesos basado en el respectivo virus biológico, los cuales son descritos a continuación [14]:

El virus infecta a un conjunto inicial de células, como se observa en la Fig. 2, las cuales replican cada virus.

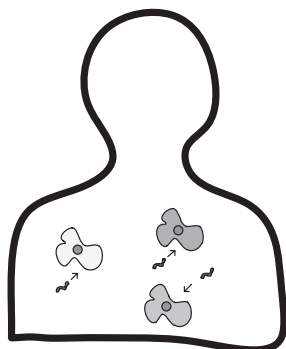


Fig. 2. Infección de Virus a Grupo Inicial. El virus infecta un conjunto inicial de células. Las cuáles a su vez replican a otras células el virus adquirido.

Los virus, con base en la fisiología de las células infectadas recientemente, determinan qué características tienen las células altamente susceptibles a la infección.

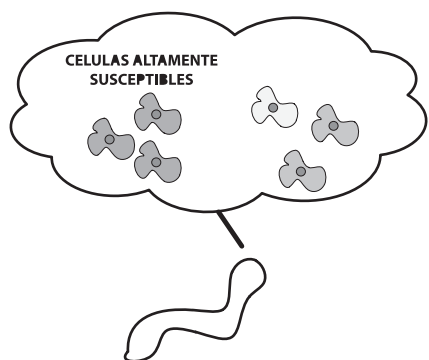


Fig. 3. Virus ataca células más susceptibles, basándose en la fisiología de las células infectadas recientemente.

Los nuevos virus atacan células altamente susceptibles y al tejido circundante. El proceso se repite hasta infectar todo el organismo. La estructura del algoritmo Ébola es la siguiente [14]:

Inicio
 El virus infecta un conjunto de células de manera aleatoria
 Se calcula la susceptibilidad o resistencia de las células a la infección
 Mientras no se cumpla el criterio de parada hacer
 Se identifican dentro de las células infectadas recientemente aquellas altamente susceptibles
 Se identifica el perfil fisiológico típico de las células altamente susceptibles usando la información de las células infectadas recientemente
 Usando la información de las células altamente susceptibles se identifica el perfil fisiológico de las nuevas células a infectar
 El virus infecta las nuevas células
 El virus infecta el tejido cercano a las nuevas células infectadas
 Fin Mientras
 Fin

En la adaptación del algoritmo Ébola al problema de enrutamiento QoS se consideró que las células serían posibles soluciones factibles del problema estudiado, es decir, su estructura es un vector fila que contiene una ruta factible. Por ejemplo, para una instancia de tamaño 20 se tendría la siguiente estructura:

1	58	...	17	20
---	----	-----	----	----

Fig. 4. Estructura de una célula. Vector fila que representa una trayectoria factible incluyendo nodo origen, nodo destino y nodos intermedios.

De tal manera que la primera posición del vector será el nodo origen, la última posición será el nodo destino y las posiciones intermedias serán nodos siempre alcanzables desde el nodo anterior.

Para el cálculo de la susceptibilidad o resistencia de las células a la infección, se utilizó el valor obtenido en cada solución para la función objetivo; en este caso, el retraso extremo a extremo. Es decir, las células altamente susceptibles a la infección, serán aquellas con menor retraso extremo a extremo.

La identificación del perfil fisiológico típico de las células altamente susceptibles, se realizó mediante la construcción de una matriz que contiene en cada fila una célula altamente susceptible, es decir, cada fila incluye una ruta que es una solución factible al problema de enrutamiento. A partir de la matriz anterior se construyó un vector fila, que contiene en la primera posición el nodo origen y en la última posición el nodo destino. Para todos los nodos intermedios se cumple que, la *i*-ésima posición del vector fila es igual a la moda de la *i*-ésima columna de la matriz de células altamente susceptibles, como se muestra en la figura 5.

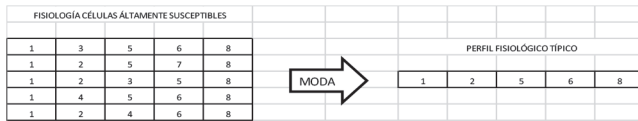


Fig. 5. Perfil fisiológico típico de células altamente susceptibles.

Una vez se obtiene el perfil fisiológico típico, se realiza una combinación de características entre éste y las células altamente susceptibles dando como resultado nuevas células, las cuales representan nuevas soluciones factibles o rutas con los menores retraso extremo a extremo en el problema de enrutamiento, como se muestra en la figura 6.

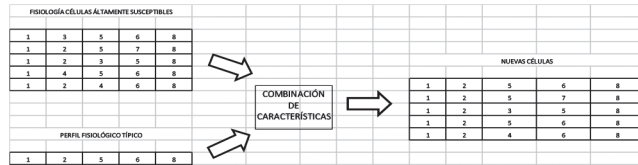


Fig. 6. Obtención de nuevas células mediante combinación de células altamente susceptibles y el perfil fisiológico típico.

Una nueva célula infectada contiene una ruta que es solución factible al problema estudiado. Para generar el tejido cercano a ésta, se construye una matriz que contiene en cada fila rutas que difieren de la nueva célula, en un solo nodo. La figura 7 muestra un ejemplo de este proceso.

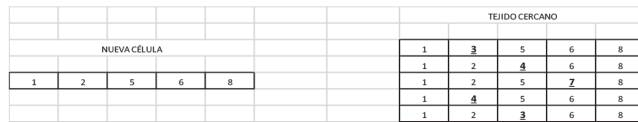


Fig. 7. Generación de tejido cercano a partir de la modificación de células.

V. METODOLOGÍA

Los algoritmos objeto de este trabajo fueron implementados en el software Matlab R2009b y comparados usando un experimento simple para el cual se calcularon siete réplicas [14].

Los resultados del experimento se analizaron de manera descriptiva para lo cual se calculó el promedio de retraso obtenido por cada algoritmo comparado, en cada una de las quince instancias resueltas. Las instancias tienen tamaños de 20, 30 y 50 nodos y fueron generadas de manera aleatoria.

De igual manera se realizó una prueba de hipótesis comparando los retrasos obtenidos por los algoritmos en la solución de las instancias.

VI. RESULTADOS

Según el análisis descriptivo del promedio de retraso, el algoritmo Ébola supera a genético en 9 de las 15 réplicas, y en una obtiene un empate en el promedio de retraso, por lo tanto el 66,66% de las veces el algoritmo Ébola obtiene

un mejor promedio de retraso que algoritmo genético. La tabla II, muestra el promedio de retraso obtenido por cada algoritmo en la solución de cada una de las instancias.

En el análisis inferencial se realizó una prueba de hipótesis no paramétrica obteniendo los siguientes resultados presentados en la tabla III:

Debido a que el valor-P de las pruebas realizadas es menor a 0,05, se puede rechazar la hipótesis de que los datos de Delay provienen de una distribución normal con 95% de confianza

Comparación de Medianas para Delay

Mediana de genético: -0,113477

Mediana de ébola: -0,267261

Prueba W de Mann-Whitney (Wilcoxon) para comparar medianas:

Hipótesis Nula: mediana genético = mediana ébola

Hipótesis Alternativa: mediana genético < mediana ébola

W = 5003,0 valor-P = 0,24749

No se rechaza la hipótesis nula para alfa = 0,05.

Debido a que el valor-P es mayor que 0,05, no hay diferencia estadísticamente significativa entre las medianas con un 95,0%.

TABLA II
PROMEDIO DEL RETRASO EN ALGORITMOS GENÉTICO Y ÉBOLA

Instancia	Número de Nodos	Descripción de Enlaces	Genético	Ébola	Promedio
1	20	UTP-5 clase A	15,85714286	11,14285714	13,5
2	20	UTP-5 clase A	8,714285714	7,428571429	8,071428571
3	50	UTP-5 clase A	8,857142857	8,285714286	8,571428571
4	20	UTP-5 clase A	12,71428571	12,71428571	12,71428571
5	80	UTP-5 clase A	8,285714286	8	8,142857143
6	50	UTP-5 clase A	17,71428571	5,285714286	11,5
7	50	UTP-5 clase A	9,714285714	10,42857143	10,07142857
8	80	UTP-5 clase A	10	10,42857143	10,21428571
9	20	UTP-5 clase A	10	12	11
10	20	UTP-5 clase A	11,71428571	10,57142857	11,14285714
11	50	UTP-5 clase A	12	4,857142857	8,428571429
12	80	UTP-5 clase A	4,285714286	8	6,142857143
13	20	UTP-5 clase A	11,42857143	11,85714286	11,64285714
14	20	UTP-5 clase A	14	9,142857143	11,57142857
15	50	UTP-5 clase A	13,14285714	16,71428571	14,92857143
Promedio			11,22857143	9,79047619	10,50952381

TABLA III
PRUEBA DE NORMALIDAD PARA DELAY

Prueba	Estadística	Valor-P
1	15,85714286	11,14285714

Los datos obtenidos por el algoritmo Ébola evidencian un mejor desempeño frente a los resultados obtenidos por el Algoritmo Genético en la solución de las instancias usadas, obteniendo un menor promedio y una menor variabilidad como se observa en la figura 8.

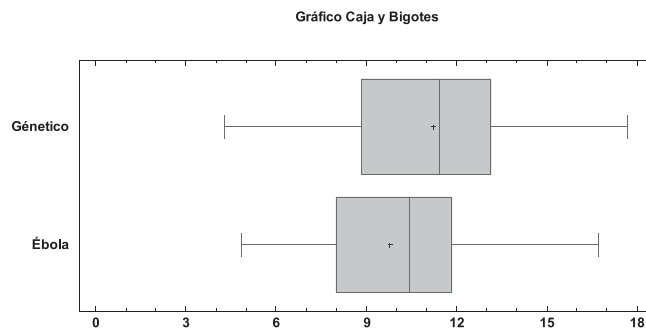


Fig. 8. Gráfico de caja y bigotes del retraso extremo a extremo.

VII. CONCLUSIONES

En la solución de las instancias evaluadas Algoritmo Genético obtuvo un promedio de 11,22 en el retraso, mientras que Ébola alcanzó 9,79. También se realizó una prueba de hipótesis no paramétrica (Prueba W de Mann-Whitney) que obtuvo un valor-P = 0,24, lo que muestra que no existen diferencias significativas entre los promedios de retraso obtenidos por los dos algoritmos comparados.

El nivel de calidad servicio está determinado por el cálculo de métricas como retardo extremo a extremo, latencia, variaciones de latencia, pérdida de paquetes y ancho de banda, entre otros. En el presente trabajo el nivel de calidad de servicio está dado por la métrica retraso extremo a extremo (*delay end to end*). El algoritmo ébola obtuvo buenos resultados utilizando esta métrica frente al algoritmo genético, lo cual implica que las soluciones obtenidas tienen buen nivel de calidad de servicio.

Lo anterior nos permite afirmar que el Algoritmo Ébola proporciona tan buenos resultados como Algoritmo Genético en la solución del Unicast Routing Problem, por lo que se convierte en una novedosa herramienta en la solución del mismo. Esta afirmación se fundamenta en el hecho de que Algoritmo Genético es una de las técnicas metaheurísticas más utilizada en la solución del problema estudiado [6].

REFERENCIAS

- [1] W.C. Lee, M.G. Hluchyj y P.A. Humblet, "Routing Subject to Quality of Services Constraints in Integrated Communications Networks," IEEE Network, 1995, July/August, pp. 46–55.
- [2] Z. Wang and J. Crowcroft, "Quality of Service Routing for Supporting

- [3] X. Sun y X. Lv, "Novel dynamic ant genetic algorithm for QoS routing in wireless mesh networks," in Proc. WiCom, 2009, China, pp. 1-4.
- [4] L. Barolli, A. Koyama, H. Sawada, T. Sukanuma, N. Shiratori, "A New QoS Routing Approach for Multimedia Applications Based on Genetic Algorithms," in Proc. CW, 2002, Japan.
- [5] G. Xue, "Minimum Cost QoS Multicast and Unicast Routing in Communication Networks," IEEE Trans. Commun., 2003, vol. 51, no. 5, pp.817-824
- [6] R. Leela, N. Thanulekshmi y S. Selvakumar, "Multi-constraint QoS Unicast Routing Using Genetic Algorithm (MURUGA)," Applied Soft Computing, 2011, vol. 11, no. 2, pp. 1753-1761.
- [7] Y. Zheng, J. Tian, Z. Liu y W. Dow, "A Limited Path Unicast QoS Routing Algorithm," in Proc. ISCC, 2004, China, pp. 870 – 875.
- [8] A. Koundinya, A. Negi, V. Sastry, "Unicast Routing Algorithm with Multiple Quality-of-Service Parameters," in Proc. ISPAN, 2004, India, pp. 520-525.
- [9] A. Jüttner, B. Szviatovski, I. Mécs, Z. Rajkó, "Lagrange Relaxation Based Method for the QoS Routing Problem," in Proc. INFOCOM, 2001, pp. 859–868.
- [10] S. Chen, y K. Nahrstedt, "An Overview of Quality of Service Routing for Next Generation High-Speed Networks: Problems and Solutions," IEEE Network, 1998, pp. 64–79.
- [11] Y. Donoso, R. Fabregat, "Multi-Objective Optimization in Computer Networks Using Metaheuristics," New York: Auerbach Publications, 2007, pp. 78-104.
- [12] D. Sadava et al, "Vida. La Ciencia de la Biología." Argentina: Editorial Médica Panamericana, 2009.
- [13] N. Vargas, C. Rincón, E. Reyes, "Formación de poro en membrana celular por medio de la pequeña glicoproteína de secreción del virus Ébola Zaire," Universitas Scientiarum, 2004, vol. 9, no. 2, pp. 43-48.
- [14] J. López, H. Madrid, "Ébola: Un nuevo método para los problemas de optimización", Trabajo fin de carrera, Dpto. Ing. Industrial, Universidad de Córdoba, Montería, 2011.
- [15] D. Montgomery, "Diseño y Análisis de Experimentos." México: Editorial Limusa, 2004.

Velsy Liliana Hernández Riaño, Ingeniera de Sistemas egresada de la Universidad Distrital Francisco José de Caldas, M.Sc. en Teleinformática de la Universidad Distrital Francisco José de Caldas. Actualmente es Profesora Asistente del Departamento de Ingeniería de Sistemas y Telecomunicaciones de la Universidad de Córdoba. Entre sus temas de investigación están: optimización de redes, calidad de servicio y TIC's aplicadas a entornos de aprendizaje.

Jorge Mario López Pereira, Ingeniero Industrial egresado de la Universidad de Córdoba, Especialista en Informática y Telemática de la Fundación Universitaria del Área Andina y M.Sc. en Ingeniería Industrial de la Universidad del Norte. Actualmente es Profesor Asistente del Departamento de Ingeniería Industrial de la Universidad de Córdoba. Sus intereses en investigación incluyen: optimización combinatoria, inteligencia artificial y enseñanza de la ingeniería.

Helman Enrique Hernández Riaño, Ingeniero Industrial egresado de la Universidad Distrital Francisco José de Caldas, Especialista en Gerencia de Producción y Calidad de la Universidad Tecnológica de Bolívar, M.Sc. en Gestión de Organizaciones de la Universidad EAN y es candidato a Doctor en Ingeniería Industrial de la Universidad del Norte. Actualmente es Profesor Asistente del Departamento de Ingeniería Industrial de la Universidad de Córdoba. Sus intereses en investigación incluyen: optimización combinatoria, inteligencia