



Herramienta software para la implementación de algoritmos basados en técnicas metaheurísticas, orientados a optimizar el establecimiento de rutas para el flujo de información en comunicaciones de multidifusión

Software tool for the implementation of algorithms based on metaheuristic techniques, aimed at optimizing the development of routes for the flow of information in multicast communications

Juan Manuel Rodríguez*

John Alejandro Benito**

Roberto Emilio Salas Ruiz***

Fecha de recepción: 20 de enero del 2011

Fecha de aceptación: 16 de junio del 2011

Resumen

En las transmisiones multicast, la optimización en el establecimiento de rutas es un área de especial interés en la comunidad académica. La necesidad de este tipo de transmisiones se pueden ver en las aplicaciones relacionadas con la telemedicina, la educación, el trabajo a la distancia, entretenimiento, etc El objetivo de este trabajo es ser capaz de crear una herramienta de software que propone una o un conjunto de rutas a seguir en el momento de hacer una transmisión multicast. Esta ruta o un conjunto de rutas son el resultado de la ejecución de los algoritmos de recocido simulado y búsqueda tabú, cuyo principal objetivo es optimizar funciones Hop Count, Delay, Cost and Bandwidth consumption.

Palabras clave: metaheurísticas, optimización multiobjetivo, redes de multidifusión, recocido simulado, búsqueda tabú.

* Universidad Distrital Francisco Jose de Caldas. Correo electrónico: juanmrodriguezr@gmail.com

** Universidad Distrital Francisco Jose de Caldas. Correo electrónico: john.insofacto@gmail.com

*** Universidad Distrital Francisco Jose de Caldas. Correo electrónico: resalasr@udistrital.edu.co

Abstract

In multicast transmissions, the optimization in the establishment of routes is an area of special interest in the academic community. The need of this type of transmissions can be seen in applications related to Telemedicine, education, work distantly, entertainment, etc. The aim of this work is be able to create a software tool that proposes one or one set of routes to follow to the moment to do a multicast transmission. This route or set of routes are the result of the execution of the algorithms Simulated Annealing and Tabu Search, which principal aim is optimize the objective's functions Hop Count, Delay, Cost and Bandwidth consumption.

General Terms: algorithms.

Key words: metaheuristics, multiobjective optimization, multicast networks, simulated annealing, tabu search.

Introducción

La demanda en trasmisiones de tipo multidifusión se define como aquella que se realiza entre uno o varios emisores y uno o varios receptores, y las cuales requieren que se planteen continuamente estrategias de mejoramiento. Aspectos como la calidad de servicio, el consumo de ancho de banda y el establecimiento de rutas de transmisión concentran la atención de la industria y la comunidad académica. El establecimiento de rutas para la transmisión de tipo multidifusión se puede modelar como un problema de naturaleza multiobjetivo; es aquí donde la aplicación de algoritmos metaheurísticos puede resultar bastante útil.

Ante lo anterior, se planteó realizar una herramienta software que proponga un conjunto de rutas por seguir, en el momento de hacer una transmisión de tipo multidifusión. Esta ruta o conjunto de rutas son el resultado de la ejecución de los algoritmos metaheurísticos *recocido simulado* y *búsqueda tabú*, cuyo objetivo principal es optimizar las funciones

objetivo *conteo de saltos*, *ancho de banda consumido*, *retardo* y *costo económico*.

El proceso para construir esta herramienta informática comenzó con la descripción teórica de los esquemas y algoritmos de encaminamiento utilizados actualmente en la transmisión de datos en redes multidifusión. Posteriormente, fue necesario determinar qué características debían tener los algoritmos *recocido simulado* y *búsqueda tabú* para operar sobre un problema multiobjetivo. Y finalmente, se estableció la técnica de búsqueda de soluciones sobre la red, representada en un grafo y la estrategia de normalización necesaria para operar matemáticamente los valores de cada función objetivo.

El presente artículo está organizado en los siguientes temas: primero se presenta un contexto de las redes de datos y de las comunicaciones de multidifusión, luego se sigue con el área de optimización y optimización multiobjetivo, para luego presentar la herramienta desarrollada.

Redes de datos

Concepto

Una red es una combinación de hardware y software que envía datos desde un sitio a otro. El hardware consta del equipo físico que transporta las señales de un punto de la red a otro. El software consta de conjuntos de instrucciones que hacen posible los servicios que se esperan de una red

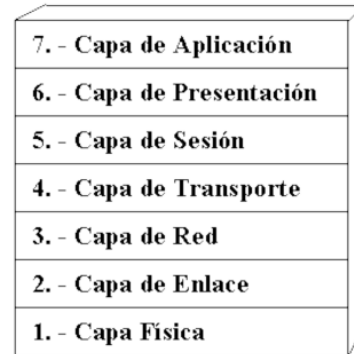
Clasificación

Existen diferentes tipos de clasificación de las redes de datos, de acuerdo con el contexto de estudio que se esté realizando. Según la topología, los esquemas más importantes son: árbol, bus, anillo, estrella o malla. De acuerdo con el tipo de transmisión, pueden ser punto a punto, punto a multipunto o de difusión. Y desde el punto de vista de su extensión pueden ser redes de área personal PAN, redes de área local LAN, redes de área metropolitana MAN y redes de área amplia WAN.

Modelo OSI

El modelo de Interconexión de Sistemas Abiertos OSI fue desarrollado por la Organización Internacional de Estandarización ISO, con el ánimo de establecer un modelo de referencia para el diseño de sistemas de red. Forouzan (2007) explica que al desarrollar el modelo, los diseñadores refinaron el proceso de transmisión de datos hasta los elementos más fundamentales. Identificaron qué funciones tienen usos relacionados y unieron todas las funciones dentro de grupos discretos que se convirtieron en niveles. El modelo está compuesto por siete niveles, que se presentan en la figura 1.

Figura 1. Capas del modelo OSI



Fuente: elaboración propia.

Modelo TCP/IP

Es la definición de una arquitectura de protocolos de red. Su desarrollo se llevó a cabo en la década de los setenta, por un programa de la agencia DARPA (Defense Advanced Research Projects Agency), perteneciente al departamento de defensa de los Estados Unidos. Las investigaciones y desarrollos fueron realizados sobre la red de conmutación de paquetes Arpanet, por esto, TCP/IP es considerada como su evolución. Está constituido por cinco capas independientes entre sí: capa de red, capa de acceso a red, capa de Internet, capa de transporte y capa de aplicación. Es de anotar que en este modelo el enrutamiento se realiza utilizando el protocolo IP (Internet Protocol). IP transporta los datos en paquetes denominados datagramas, cada uno de los cuales se transporta de forma independiente.

Tipos de transmisión

Los tipos de transmisión que pueden existir en una red son los siguientes:

- Transmisión unidestino: es la realizada entre exactamente un emisor y un recep-

tor, es decir, que existe una relación uno a uno entre los dispositivos involucrados.

- Transmisión de difusión: es ejecutada en un único proceso desde un solo emisor a todas las máquinas conectadas a una red. Este tipo de difusión no requiere realizar ningún tipo de copia en el origen para enviar el datagrama IP a varios destinatarios.
- Transmisión de multidifusión: es el tipo de transmisión entre un emisor y varios receptores, estableciendo una relación de uno a muchos. Las unidades de datos o datagramas IP son enviados desde la máquina origen, a un grupo de destinatarios, también conocido como grupo de multidifusión; este grupo es identificado mediante una dirección IP de multidifusión, los destinatarios pueden o no estar en diferentes redes dispersas geográficamente.

Para poder realizar la transmisión desde un origen a varios destinos, es necesaria la implementación de algoritmos de búsqueda para calcular las mejores rutas a todos los miembros del grupo. Estos algoritmos requieren de una representación tipo árbol construida desde el origen de la información, hasta cada uno de sus destinos. Esta transmisión se puede hacer bajo dos esquemas:

- Esquema centralizado: las decisiones de encaminamiento son tomadas por un único dispositivo de encaminamiento, denominado núcleo central. Este dispositivo es el encargado de construir el árbol para todos los demás nodos de la red y ejecutar el algoritmo de búsqueda. La principal característica de este enfoque es que el núcleo, tiene toda la información referente a la red.
- Esquema distribuido: es el utilizado en la mayoría de redes actualmente, ya que cada dispositivo de encaminamiento es el encargado de construir el árbol a los destinos y ejecutar el algoritmo de búsqueda.

Protocolos de encaminamiento en esquemas distribuidos

Los tipos de transmisión:

- Encaminamiento basado en el vector distancia: los protocolos de este tipo son generalmente utilizados en redes pequeñas. Su característica principal es que utilizan el algoritmo de Bellman-ford para calcular el camino de menor costo. El algoritmo utiliza como fuente de datos las tablas de encaminamiento que son el resultado del intercambio de información entre los encaminadores. Algunos protocolos que utilizan este tipo de encaminamiento son: RIP (Routing Information Protocol) protocolo de información de ruteo, IGPR (Interior Gateway Routing Protocol), entre otros.
- Encaminamiento basado en el estado del enlace: para encaminamiento multienvío, un nodo necesita revisar la interpretación de estado. Un nodo anuncia sobre el enlace a cada grupo que tiene cualquier miembro. Aquí el significado de estado es "qué grupos están activos en este enlace". Una vez el dispositivo conoce el estado de los demás nodos, es capaz de crear el árbol de expansión utilizando el algoritmo de Dijkstra. Los protocolos que usan este tipo de encaminamiento crean por demanda un árbol o mapa por cada grupo de destino. Los protocolos más destacados son: OSPF (Open Shortest Path First): protocolo abierto del primer camino más corto; y MOSPF (Multicast Open Shortest Path First): protocolo de multidifusión abierto del primer camino más corto.

Optimización

La optimización constituye una clase específica de problemas. En términos generales, dicho problema es una pregunta cuya respuesta es una función de varios parámetros. La

optimización consiste en encontrar la mejor solución posible a un problema; dichos problemas, en el contexto de su aplicación (teórica o práctica), a que el espectro de soluciones sea más o menos difícil de encontrar. En consecuencia, la forma como se busque será el factor que determine su exactitud y su eficiencia. Usualmente, una solución a un problema de optimización requiere de un algoritmo configurable, el cual, cuando se aplica a una instancia del problema, produce la solución que se quiere.

Métodos de optimización

Para el planteamiento y la solución de problemas de optimización hay varios métodos, que son utilizados en función de varios factores y con diferentes fines.

- Algoritmos exactos: el algoritmo que busca solucionar el problema se ejecuta en un tiempo polinomial. Generalmente, se utiliza un algoritmo polinomial, aunque no necesariamente se deba utilizar uno de estos algoritmos en todos los casos de este tipo. Se dice que estos problemas pertenecen a la clase P, porque se dispone de un algoritmo que tarda en resolverlo mediante un polinomio $p(n)$, que depende del tamaño n del problema; de manera que al crecer el tamaño del problema, el tiempo que tarda el algoritmo en resolver el problema crece, pero despacio (polinómicamente), ya que la velocidad de crecimiento de un polinomio es lenta.
- Algoritmos de aproximación: generalmente, se aplica a problemas duros (hard). Los problemas duros se refieren a problemas que no son fáciles computacionalmente, es decir, no se ha descubierto un algoritmo de tiempo polinomial para poder solucionarlos. Entre este tipo de algoritmos se puede hacer una división, entre algoritmos heurísticos y metaheurísticos.

Algoritmos heurísticos

Son los más fáciles de utilizar, ya que se basan en el conocimiento de una heurística que guía el proceso de búsqueda. Por lo general, el conocimiento del problema ayuda a encontrar una heurística razonable que encontrará rápidamente una solución aceptable. Un algoritmo de este tipo solo buscará dentro de un subespacio del área total a una solución buena –que no necesariamente es la mejor– que satisfaga las restricciones impuestas. La principal limitación es su incapacidad para escapar de óptimos locales –encontrar soluciones parcialmente óptimas–. Los algoritmos heurísticos más conocidos son el algoritmo voraz y búsqueda local.

Algoritmos metaheurísticos

Una metaheurística es un proceso iterativo maestro que guía y modifica las operaciones de una heurística subordinada para producir eficientemente soluciones de alta calidad. Las metaheurísticas pueden manipular una única solución completa (o incompleta) o una colección de soluciones en cada iteración. La heurística subordinada puede ser un procedimiento de alto (o bajo) nivel, una búsqueda local, o un método constructivo. Entre los algoritmos metaheurísticos más conocidos están, el recocido simulado, búsqueda tabú y los algoritmos genéticos.

Metaheurística de recocido simulado

Para superar el problema que puede ocurrir con métodos como búsqueda local –que puede quedar atrapado en óptimos locales–, se introduce el método de recocido simulado que permite empeoramientos en la solución mediante reglas de probabilidad que aceptan dichas soluciones. El nombre proviene del símil que se hace con el proceso meta-

lúrgico de recocido, el cual, por medio de un enfriamiento lento, consigue un sólido más resistente.

Al principio, esta estrategia acepta casi todas las soluciones, lo que permite explorar todo el conjunto factible; después, gradualmente se disminuye la permisividad –llamada temperatura–, de manera que la aceptación de movimientos es más selectiva cada vez, hasta que finalmente, solo se aceptan soluciones que mejoren la solución actual. El algoritmo se compone de varios componentes, a saber:

- **Temperatura inicial:** la temperatura del sistema se denomina T , que tenderá a reducirse durante el proceso de optimización. Si se selecciona un valor muy bajo, el algoritmo no posibilita empeoramiento; por el contrario, si se selecciona un valor muy alto, el proceso se hace lento.
- **Esquema de enfriamiento:** si decrece muy rápido, puede quedar atrapado en un óptimo local y si decrece muy lento, también se ejecuta muy lento. Lo que generalmente se hace es mantener la temperatura constante durante L iteraciones, luego se reduce multiplicándose con (entre 0 y 1). El valor típico de α suele ser 0,95.
- **Condición de equilibrio:** el sistema utiliza cadenas de Markov en la generación de las sucesiones (vecindario).
- **Criterios de parada:** se usan habitualmente dos reglas de parada: parar si la función no ha mejorado en al menos α % tras k_1 iteraciones. Parar si el número de transiciones aceptadas es menor que el β % de L tras las k_2 iteraciones.

Metaheurística de búsqueda tabú

La búsqueda tabú se basa en la exploración en vecindarios evitando los óptimos locales, pero en una forma determinística, la cual trata de modelar los procesos de la memoria humana. La búsqueda tabú incorpora me-

moria adaptativa y exploración sensible. La memoria adaptativa es el mecanismo por el cual la búsqueda tabú asegura que las soluciones ya se encuentran en pasos anteriores para no repetirlos. El mecanismo almacena las soluciones de pasos ya tomados en una memoria temporal.

La idea básica es permitir el paso de una solución a otra (movimiento), así empeore el objetivo. Para evitar que se quede en óptimos locales se usa la lista tabú. En ella, se guarda durante cierto tiempo un atributo que permite identificar la solución o el movimiento realizado, de manera que todo movimiento que tenga un atributo en la lista tabú, se considera prohibido y no se puede realizar. Algunos aspectos para tener en cuenta en el método son:

- **Elección de los atributos:** es importante la elección de los atributos y un movimiento puede tener atributos múltiples. No puede ser muy general –podría permitir pocos movimientos– o muy específico –puede impedir la salida de una región con mínimos locales.
- **Longitud de la lista tabú:** permite que los atributos tabú estén presentes durante cierto tiempo, antes de que sean eliminados de allí y se pueda volver a permitir su utilización.
- **Elección de la regla de parada:** usualmente, se suele fijar a priori el número de iteraciones.
- **Criterio de aspiración:** es un valor de referencia con el cual se permite que una solución que tiene atributos tabú pueda ser seleccionada como la solución actual, anulando la prohibición

Optimización multiobjetivo

Cuando se habla de optimización simple, o de objetivo simple, quiere decir que la fun-

ción por maximizar o minimizar es una sola, de manera que cuando se encuentre dicho óptimo (o se aproxime a él), ya estará solucionado el problema. Cuando se habla de optimización multiobjetivo, hace referencia a que hay más de una función por maximizar (o minimizar) y, por tanto, la solución consistirá de un conjunto de valores que optimicen cada uno de los objetivos. Dicho conjunto de soluciones se conoce como óptimo de Pareto, el cual se plantea como la situación en la que no se puede mejorar una parte (función) de la solución sin perjudicar alguna de las otras partes.

- Suma ponderada: el método consiste en convertir todas las funciones en una sola, ponderándolas con la asignación de un peso a cada función. Cada función se multiplica por un peso (w_i) cuyo valor está entre 0 y 1. La suma de los pesos debe dar 1. El resultado es una combinación lineal de las funciones.

$$F'(X) = \sum_{i=1}^n r_i * f_i(X) \quad (1)$$

- Método e-constraint: consiste en crear un modelo de objetivo simple en el que una de las funciones será optimizada y las demás se tratarán como restricciones en el modelo. A cada una de las funciones que se va a tratar como restricción se la somete a un límite dado por una constante k , de manera que cambiando los valores de k para cada una de las nuevas restricciones se obtenga diferentes resultados de optimización.

$$f_i(X) \leq \varepsilon_k, k = 1, \dots, n \text{ y } k \neq i \quad (2)$$

Herramienta de software desarrollada

Análisis del problema

En esta etapa, se establece el modelo por optimizar, especificando el dominio del proble-

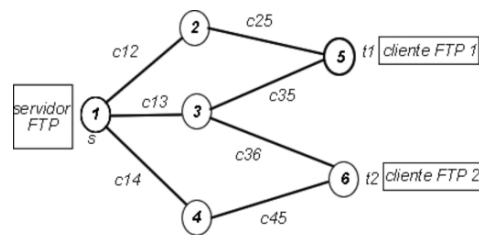
ma combinatorio y la propuesta dentro del marco de los algoritmos metaheurísticos.

Modelamiento de la red

En una red de computadores (como por ejemplo red local), el modelamiento de esta puede ser representada por medio de un grafo, en el cual los vértices se pueden identificar como los equipos de comunicaciones dispuestos para el tránsito de información –como routers o switches–, y las aristas son las líneas físicas –como cable coaxial, fibra óptica o microondas– que comunican dichos vértices.

En ese sentido, se puede decir que una red es un grafo $G = (N, E)$, donde N es el conjunto de vértices y E son las aristas que los unen. Dentro de dicho grafo, hay vértices origen (denotados como s) y conjuntos de vértices destino (denotados con T), un vértice destino t_f pertenece a un flujo f . El conjunto de flujos para ser transmitidos por la red se denotan con F , y un flujo f pertenece a F . Asimismo, T_f es el conjunto de vértices destino para el flujo f . El costo que tiene transmitir un flujo entre dos vértices i y j se denomina c_{ij} . La transmisión de un flujo f desde un origen s hasta un conjunto de destinos T , tiene una demanda de tráfico, denominada bw_f . En la figura 2, se puede ver el esquema de un grafo para la transmisión de un flujo de datos FTP.

Figura 2. Grafo para la transmisión de un flujo de datos FTP



Fuente: <http://es.scribd.com/doc/50828699/9/Protocolo-de-Control-de-Transmision-TCP>

En una transmisión de Internet TCP/IP –que son las más ampliamente difundidas entre redes locales e Internet–, los flujos de información son generados por aplicaciones –como transferencia de archivos, de hipertexto, entre otros– y transmitidos por IP entre el vértice

origen y los vértices destino. El encaminamiento entre los vértices intermedios es lo que se pretende determinar de manera óptima para los diferentes factores para tener en cuenta en el modelamiento del problema. En la tabla 1, están los parámetros indicados.

Tabla 1. Parámetros en un grafo de flujo de información

Expresión	Definición
$G(N, E)$	El grafo que representa la red.
N	Conjunto de nodos.
E	Conjunto de aristas.
S	El vértice origen.
T	Conjunto de vértices destino.
(i, j)	Arista del vértice i a j .
F	El conjunto de flujos.
f	Cualquier flujo de multidifusión.
T_f	El subconjunto de vértices destino para el flujo f .
c_{ij}	La capacidad disponible de la arista (i, j) .
bw_f	La demanda de tráfico del un flujo f .

Fuente: <http://es.scribd.com/doc/50828699/9/Protocolo-de-Control-de-Transmision-TCP>

Modelamiento del tráfico multidifusión

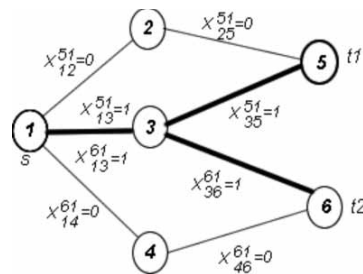
El modelo se plantea a partir de una variable que indica la ruta a través de la cual el flujo

de información será transmitido. La variable es X_{ij}^f , la cual se define como:

$$X_{ij}^f = \begin{cases} 1, & \text{si la arista } (i, j) \text{ es usada por el flujo } f \text{ con destino } t \\ 0, & \text{si la arista } (i, j) \text{ no es usada por el flujo con destino } t \end{cases} \quad (3)$$

De esta manera, si es utilizada una arista dentro del flujo, se sumará una al total de aristas que constituyen la ruta entre el origen y los destinos; si la arista no es utilizada –porque no constituye la ruta, o porque no hay una arista que una dichos vértices– no sumará al total de aristas usadas en la ruta. En la figura 3, se puede ver que la ruta utilizada (aristas sombreadas) está compuesta de aristas cuyo valor de la variable X es uno; las demás aristas que no son utilizadas tiene un cero.

Figura 3. Ruta utilizada en un flujo



Fuente: <http://es.scribd.com/doc/50828699/9/Protocolo-de-Control-de-Transmision-TCP>

Funciones por optimizar

Conteo de saltos

Representa la cantidad de aristas por las cuales tienen que pasar los paquetes desde el vértice de origen hasta los vértices destino. El protocolo de red que se encarga de esto es el Protocolo de Enrutamiento Multicast de Vector Distancia (DVMRP). Matemáticamente se expresa de la siguiente forma:

$$f_1 = \min \sum_{f \in F} \sum_{t \in T_f} \sum_{(i,j) \in E} X_{ij}^f \quad (4)$$

Donde:

$\sum_{(i,j) \in E} X_{ij}^f$	Es una de las posibles rutas para el flujo f que va para t .
$\sum_{t \in T_f}$	Denota cada uno de los vértices destino t del flujo f .
$\sum_{f \in F}$	Son todos los flujos de multidifusión transmitidos sobre la red.

Retardo

Hace referencia al retardo de propagación, es decir, al tiempo que le toma a un paquete atravesar las aristas intermedias antes de llegar al vértice destino. La unidad de medida son los milisegundos (ms). Para cada arista, se estima un tiempo de retardo para atravesarlo, de manera que el retardo de propagación total del envío de un paquete a sus destinos consistirá en la suma de los retardos de las aristas visitadas. La expresión matemática de esta función es:

$$f_2 = \min \sum_{f \in F} \sum_{t \in T_f} \sum_{(i,j) \in E} d_{ij} \cdot X_{ij}^f \quad (5)$$

Donde:

$\sum_{(i,j) \in E} d_{ij} \cdot X_{ij}^f$	Es el retardo de propagación de las rutas para el flujo f que va para t .
$\sum_{t \in T_f}$	Denota cada uno de los vértices destino t del flujo f .
$\sum_{f \in F}$	Son todos los flujos de multidifusión transmitidos sobre la red.

Costo

Está asociada con el costo monetario (la unidad de medida es en dinero) de usar un enlace de algún tipo de tecnología o protocolo –como podría ser el costo de usar protocolos punto a punto, usar una red ATM o una red Frame Relay– entre las aristas intermedias en una transmisión de multidifusión. En esta función se debe tener en cuenta que si un flujo es enviado a varios destinos y ellos comparten enlaces, solamente se transmite una sola vez a través de dichos enlaces; esto es así porque el vértice (enrutador) hace una copia del paquete por transmitir y lo reenvía a los destinos indicados sin necesidad de duplicar la recepción del paquete. La función matemática asociada es la siguiente:

$$f_3 = \min \sum_{f \in F} \sum_{(i,j) \in E} w_{ij} \cdot \max(X_{ij}^f)_{t \in T_f} \quad (6)$$

Donde:

$\max(X_{ij}^f)_{t \in T_f}$	Permite que solamente se cuente una vez el enlace para cualquier vértice destino t .
------------------------------	--

Consumo de ancho de banda

Hace referencia al ancho de banda que es utilizado -consumido, no el que queda disponible- en cada enlace para hacer la transmisión. La unidad de medida son los bits por segundo (bps). Aquí se tiene en cuenta la demanda de ancho de banda del flujo de información -que es constante- y con base en esa demanda, se calcula el consumo en la ruta de transmisión. Cabe anotar que si un enlace no tiene la capacidad para alojar la transmisión, no se puede usar como parte de la ruta de transmisión. En la función de consumo de ancho de banda para multidifusión, se debe tener en cuenta que si un flujo es enviado a varios destinos y ellos comparten enlaces, solamente se transmite una sola a la vez a través de dichos enlaces; esto es así porque el vértice (enrutador) hace una copia del paquete por transmitir y lo reenvía a los destinos indicados sin necesidad de duplicar la recepción del paquete. Esto permite que se optimice aún más la transmisión. La función matemática que expresa la función y muestra la evitación de envíos redundantes es la siguiente:

$$f_4 = \min \sum_{f \in F} \sum_{(i,j) \in E} b_{f,i,j} \cdot \max(X_{ij}^f)_{t \in T_f} \quad (7)$$

Donde:

$\max(X_{ij}^f)_{t \in T_f}$	Permite que solamente se cuente una vez el enlace para cualquier vértice destino t .
------------------------------	--

Restricciones del modelo

Vértice origen: para cada destino t que pertenece a T_f y para cada flujo f , solamente una ruta debe iniciar desde el vértice origen s . La expresión matemática es:

$$c1 = \sum_{(i,j) \in E} X_{ij}^f = 1, t \in T_f, f \in F, i = s \quad (8)$$

Vértice destino: para cada flujo f , solamente una ruta debe llegar a cada vértice destino t . La expresión matemática es:

$$c2 = \sum_{(i,j) \in E} X_{ij}^f = -1, i = t, t \in T_f, f \in F \quad (9)$$

Vértices intermedios: para cada flujo f , todos los vértices intermedios de ingreso tienen un vértice de salida. Si se suman los vértices entrantes y se le restan los vértices de salida en los nodos intermedios, el valor debe ser cero para todos los flujos. La expresión matemática es:

$$c3 = \sum_{(i,j) \in E} X_{ij}^f - \sum_{(j,i) \in E} X_{ji}^f = 0, t \in T_f, f \in F, i \neq s, i \notin T_f \quad (10)$$

Capacidad del enlace: evita que se envíe una demanda de tráfico más grande que la capacidad del enlace. La expresión matemática es:

$$c4 = \sum_{f \in F} b_{f,i,j} \cdot \max(X_{ij}^f)_{t \in T_f} \leq c_{ij}, (i,j) \in E \quad (11)$$

Donde:

$\sum_{f \in F} b_{f,i,j} \cdot \max(X_{ij}^f)_{t \in T_f}$	Representa la demanda del flujo para todos los flujos que son transmitidos en los enlaces. Esto no permite que una demanda de flujo exceda la capacidad del enlace.
---	---

Propuesta de optimización para la herramienta de software

El tipo de problema propuesto corresponde a la categoría NP-Hard, ya que es un problema de computación de costo mínimo en el árbol para un grupo multidifusión (NP-completo) y, además, incluye variables reales y enteros constantes. El modelo propuesto para el pro-

totipo de software contempla la optimización de las siguientes funciones objetivos:

- Conteo de saltos.
- Retardo de propagación.
- Costo.
- Consumo de ancho de banda.

Sujeto a las siguientes restricciones:

- Vértice origen.
- Vértice destino.
- Vértices intermedios.
- Capacidad del enlace.

Utilizando como método de optimización multiobjetivo a la suma ponderada. Para esto se especifica el problema de optimización multiobjetivo para multidifusión así:

$$\begin{aligned} \min Z = & r_1 \cdot \sum_{f \in F} \sum_{i \in T_f} \sum_{(i,j) \in E} X_{ij}^f + r_2 \cdot \sum_{f \in F} \sum_{i \in T_f} \sum_{(i,j) \in E} d_j \cdot X_j^f \\ & + r_3 \cdot \sum_{f \in F} \sum_{(i,j) \in E} w_{ij} \cdot \max(X_j^f)_{i \in T_f} + r_4 \cdot \sum_{f \in F} \sum_{(i,j) \in E} w_f \cdot \max(X_j^f)_{i \in T_f} \end{aligned} \quad (12)$$

Debido a que las funciones tienen distintas escalas de valores y no son congruentes entre ellas, es necesario aplicar una normalización de los valores antes de aplicar el método de suma ponderada para el cálculo de la función multiobjetivo. La implementación se hizo utilizando las metaheurísticas trayectoriales de *recocido simulado* y *búsqueda tabú*.

Arquitectura de desarrollo de la herramienta

La herramienta que se seleccionó para el desarrollo del prototipo de software es Java. Las razones para la elección de esta herramienta con respecto a otras son:

- Alto nivel: es un lenguaje con el cual se pueden desarrollar aplicaciones robustas y extensibles para casi todos los entornos de programación actuales (aplicaciones de escritorio, aplicaciones Web, móviles,

para acceso remoto, para trabajo en paralelo, entre otras).

- Orientación a objetos: manejo de programación orientada a objetos, lo que permite la reutilización de componentes, herencia, programación genérica, modelamiento de alto nivel, entre otras.
- Independencia de la plataforma: con el uso de un intérprete de comandos, se puede ejecutar una aplicación independiente de la arquitectura de hardware y del sistema operativo.
- Licencia: la licencia es GNU GPL (Licencia Pública General de GNU), lo que permite su utilización para cualquier fin, sin tener que adquirir una licencia para desarrollar aplicativos o para venderlos.
- Soporte: en Internet, hay un amplio y continuo soporte de parte de desarrolladores, lo que permite solucionar muchos de los problemas propios del desarrollo de software.
- Componentes disponibles: se dispone de la posibilidad de utilizar componentes y librerías externos, para ampliar las funcionalidades.

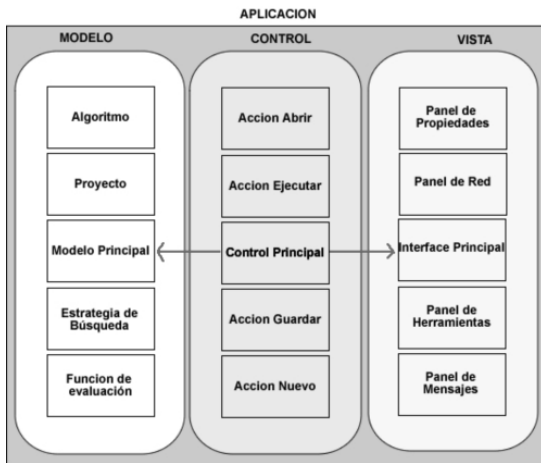
El modelo de diseño estará guiado por el patrón Modelo Vista Controlador, que hace una separación en tres niveles:

- Modelo: almacena todas las reglas y procesos relacionados con la ejecución de los algoritmos, como la forma de búsqueda de soluciones, el grafo que representa la red, las funciones objetivo, entre otros. Maneja el almacenamiento de los resultados parciales de la ejecución de los algoritmos. Manipular los archivos XML (importación y exportación de proyectos).
- Vista: contiene los componentes que construyen la interfaz de usuario (ventana principal, paneles y menús). Captura los eventos y solicitudes del usuario. Gestiona la disponibilidad de los paneles

- y las opciones de menú. Permite la creación o modificación de los parámetros de funcionamiento del aplicativo (diseño de red, parámetros del problema, parámetros de ejecución de los algoritmos).
- **Controlador:** intercepta los eventos y las solicitudes de la vista, haciendo el llamado respectivo a los componentes del modelo. Entrega objetos de datos a la vista para que se construyan los datos de las interfaces. Responde a la vista, solicitando acciones de actualización de interfaces o bloqueo de componentes de la interfaz. Sirve de puente intermedio entre la vista y el modelo; esto permite que se pueda alterar alguno de ellos sin que eso implique cambios en la otra capa.

Una muestra del modelo se ve en la figura 4.

Figura 4. Modelo vista controlador



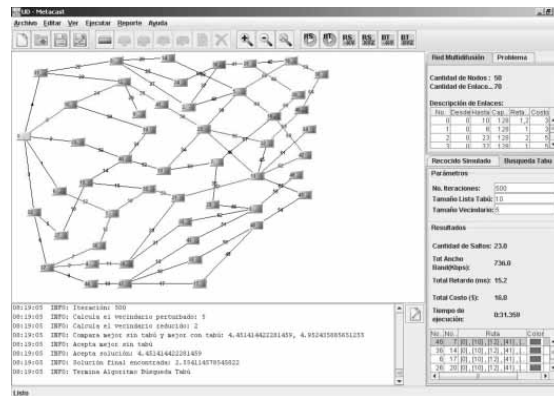
Fuente: <http://es.scribd.com/doc/50828699/9/Protocolo-de-Control-de-Transmision-TCP>

Resultados y pruebas

La herramienta fue desarrollada como una aplicación java swing. Desde esta aplicación se puede diseñar una red de datos de manera visual, así como el establecimiento de los pa-

rámetros de los algoritmos por resolver y su posterior ejecución del proceso de optimización para obtener resultados comparables de la efectividad del algoritmo. Una imagen de la vista principal de la herramienta se muestra en la figura 5.

Figura 5. Vista principal de la herramienta



Fuente: elaboración propia.

Para las pruebas de la herramienta, se dispuso de tres condiciones (redes de 15, 50 y 200 nodos), para los cuales se generaron cuatro casos de prueba (5, 8, 10 y 12 nodos destino). Los resultados de la ejecución de los casos de prueba se pueden visualizar mediante el gráfico de la red, en el que se resaltan los enlaces que corresponden con la ruta construida por el algoritmo, o en el panel de resultados, con los detalles del valor encontrado para cada una de las funciones objetivo y el valor de la función multiobjetivo (pareto). La figura 6 muestra un pantallazo de resultados que genera la herramienta (ver fig. 6).

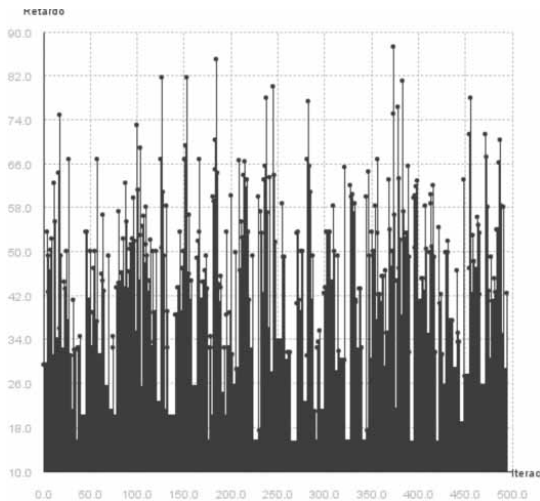
Además, se dispone de unos reportes estadísticos que muestran la evolución del proceso de optimización a lo largo de su ejecución. Dichos reportes se agrupan en reportes por objetivo, en los que se muestra una de las funciones objetivo y su evolución a lo largo de las iteraciones, tal y como se muestra en la figura 7.

Figura 6. Resultados generados herramienta



Fuente: elaboración propia.

Figura 7. Grafico de la salida de una de las funciones objetivos por optimizar

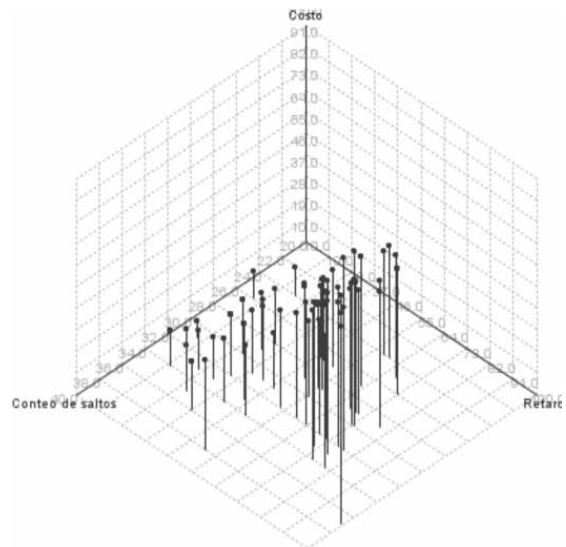


Fuente: elaboración propia.

La otra forma de ver los resultados, es mediante reportes multiobjetivo, en los que se muestran los valores de tres de las cuatro

funciones objetivo, con la posibilidad de escoger cuáles visualizar, como se muestra la figura 8.

Figura 8. Grafico de la salida de tres de las funciones objetivos por optimizar



Fuente: elaboración propia.

Conclusiones

El desarrollo del prototipo deja en evidencia la importancia que tienen las metaheurísticas en procesos de optimización, sobre todo, en ambientes como el que plantea el proyecto, en el cual el tamaño y las funcionalidades de las redes, no permiten que se pueda generar una solución genérica para todas. Las metaheurísticas que se implementaron, por su naturaleza de estrategias de búsqueda local, han mostrado sus ventajas y, por supuesto, sus desventajas, al tratarse de estrategias que tratan de escapar de soluciones falsas o comúnmente llamadas locales; esto provoca que la carga necesaria de transacciones para el establecimiento de estados dentro del proceso pueda llegar a ser desgastante o recurrente –en el caso de que no se maneje memoria.

El desarrollo de la herramienta se realizó siguiendo las directrices del patrón de diseño Modelo Vista Controlador. Este patrón resulta bastante útil, ya que facilitó la construcción inicialmente del modelo y, posteriormente, de la vista siguiendo la comunicación con el controlador. Esto permitirá adicionar nuevas funciones a la vista sin alterar el modelo y robustecer el modelo realizando cambios específicos sobre la vista.

Para trabajos futuros, se recomienda planear la arquitectura de la herramienta en términos cliente-servidor. Esto podría mejorar el desempeño en la ejecución de los algoritmos, al igual que permitir la evolución por separado tanto del cliente como del servidor. Igualmente, es posible que, a manera de aplicación real, se pueda generar un proyecto que derive de este, para que se implemente un protocolo de red local que use estas metaheurísticas.

Referencias

- Behrouz, F. (2007). *Transmisión de datos y redes de comunicación* (4ª ed.). Madrid: Mc Graw Hill.
- Chapra, S. y Canale, R. (1999). *Métodos numéricos para ingenieros* (3ª ed.). Mexico: Mc Graw Hill.
- Donoso, Y. y Fabregat, R. (2007). *Multi-Objective Optimization in Computer Networks Using Metaheuristics*. New York: Auerbach Publications.
- Duarte, A., Pantrigo, J.J. y Gallego, M. (2007). *Metaheurísticas*. Madrid: Universidad Rey Juan Carlos.
- Durillo, J.J. y Nebro, A. (2006). *jMetal: A java Framework for developing Multi-objective Optimization Metaheuristics*. Recuperado de: <http://jmetal.sourceforge.net>
- Escolano, F. et ál. (2003). *Inteligencia Artificial (Modelos, técnicas y áreas de aplicación)*. Madrid: Thompson.
- Forouzan, B. (2007). *Transmisión de datos y redes de comunicaciones* (4ª ed.). Madrid: Mc Graw Hill.
- Gacitua, D. (2007). *Evaluación de algoritmos de ruteo multipunto en redes de computadores*. Monografía, Santiago de Chile: Universidad de **Santiago de Chile**.
- Goberna, M. et ál. (2004). *Optimización lineal. Teoría, métodos y modelos*. Madrid: Mc Graw Hill.
- Michiels, W. y Aarts, E. (2007). *Theoretical Aspects of Local Search*. Berlin: Springer.
- Rusell, S. (2004). *Inteligencia Artificial. Un enfoque moderno* (2ª ed.). Madrid: Prentice Hall.
- Sait, S. y Youssef, H. (1999). *Iterative Computer Algorithms with Applications in Engineering*. California: Ed. IEEE Computer Society.