

Herramienta 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

Tool for the implementation of algorithms based on metaheuristic techniques aimed at optimizing the establishment of paths for the flow of information in multicast communications

John A. Benito
Indra Sistemas
jabenitog@indra.es

Juan M. Rodríguez
Ingeniería de Software Bancario Isban
jrodriguezro@isban.cl

En transmisiones de tipo multidifusión, la optimización en el establecimiento de rutas es un área de especial interés para la comunidad académica. La necesidad de este tipo de transmisiones se puede ver en aplicaciones relacionadas con la telemedicina, la educación, el trabajo a distancia, el entretenimiento, entre otras. El propósito de este trabajo consiste en realizar una herramienta software que proponga una o un conjunto de rutas a seguir al 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 de propagación y Costo Económico.

Palabras clave: metaheurística, multidifusión, optimización, transmisión

In multicast transmissions, optimization in the establishment of paths is an area of special interest to the academic community. The need for this type of transfer can be seen as telemedicine, education, remote work, entertainment, and other applications. The purpose of this work is to perform a software tool to propose one or a set of routes to follow when making a multicast transmission. This route or set of routes are the result of the implementation of metaheuristic algorithms, simulated annealing and tabu search, the main objective is to optimize the objective functions Hops Count, Bandwidth Consumed, Propagation Delay and Economic Cost.

Keywords: metaheuristic, multicast, optimization, transmission

Introducción

La demanda en transmisiones de tipo multidifusión, definida esta como aquella realizada entre uno o varios emisores y uno o varios receptores, requiere que sean planteadas continuamente estrategias de mejoramiento. Aspectos como la calidad de servicio, el consumo de ancho de banda y el establecimiento de rutas de transmisión, concentran el interés de la industria y la comunidad académica. El establecimiento de rutas para la transmisión de tipo multidifusión puede modelarse como un problema de naturaleza multiobjetivo, y es aquí donde la aplicación de algoritmos metaheurísticos puede resultar bastante útil.

El propósito de este trabajo consistió en realizar una herramienta software para la realización de una o un conjunto de rutas a seguir, al momento de hacer una transmisión de tipo

Fecha recepción del manuscrito: Agosto 10, 2011
Fecha aceptación del manuscrito: Octubre 30, 2011

John A. Benito, Indra Sistemas; Juan M. Rodríguez, Ingeniería de Software Bancario Isban.

Esta investigación fue financiada por: Universidad Distrital Francisco José de Caldas.

Correspondencia en relación al artículo debe ser enviada a: John A. Benito. Email: jabenitog@indra.es

multidifusión. Esta ruta o conjunto de rutas fueron 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 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 artículo está organizado en cuatro temas. En el tema inicial se describe y plantea el problema; también se definen los objetivos de los procesos de optimización y se detalla la base teórica de los temas tratados. El moldeamiento del problema y la descripción de los componentes de diseño son tratados en el segundo. El tercer tema define los lineamientos de codificación y la distribución de las piezas de software y finalmente en la cuarta parte, damos cuenta de la realización de las pruebas de sistema de la herramienta.

Optimización de recursos utilizando modelos multiobjetivo

Este artículo parte de la pregunta problema ¿Se puede optimizar la transmisión de datos en redes de multidifusión, aplicando algoritmos basados en técnicas metaheurísticas? Para responderla es necesario considera los siguientes aspectos.

Históricamente, en el contexto de la transmisión de información, las comunicaciones han evolucionado sobre dos modelos distintos, la comunicación punto a punto, como la telefonía y la comunicación de difusión, como es el caso de la televisión. El progresivo auge de internet y las redes de datos han propiciado un incremento significativo en el uso del tipo de transmisión multidifusión o de multidifusión. Este tipo de transmisión responde a las necesidades de comunicación de un grupo de usuarios conformado por uno o varios emisores y uno o varios receptores. El modelo de multidifusión incide principalmente en aplicaciones relacionadas con la telemedicina, la educación, el trabajo a distancia, el entretenimiento, entre otras.

El esquema de una red de datos se puede representar mediante un modelo matemático conocido como grafo, compuesto de nodos y aristas. Todos los dispositivos interconectados a la red, como los enrutadores, conmutadores, y demás, pueden ser representados por nodos. Las aristas simbolizan las conexiones físicas entre los dispositivos tales como cables de par trenzado, fibra óptica, enlaces inalámbricos etc. Este tipo de representaciones permite la imple-

mentación de distintas estrategias algorítmicas para solucionar problemas complejos como la optimización de recursos utilizando modelos multiobjetivo.

El objetivo principal de los procesos de optimización es realizar alteraciones sobre un proceso para mejorar su rendimiento. Generalmente, los procesos de optimización son tratados para solucionar problemas de objetivo simple, esto quiere decir, que una sola función será minimizada o maximizada. Si se quiere realizar un estudio sobre problemas del mundo real, muy probablemente se necesitará optimizar un problema de naturaleza multiobjetivo, donde se requiera encontrar un conjunto de soluciones que minimicen o maximicen todas las funciones. Una característica que incrementa la complejidad de este tipo de problemas es que sus funciones objetivo entran en conflicto, en otras palabras, pues cuando se optimiza una función, la otra u otras se degradan.

Existen distintas estrategias computacionales para solucionar problemas de optimización multiobjetivo: a través de la ejecución de modelos matemáticos por medio de simulación, utilizando técnicas heurísticas o, por medio de metaheurísticas. Las técnicas metaheurísticas tienen como función principal guiar a otras heurísticas o algoritmos a encontrar el o los valores óptimos dentro del espacio de soluciones confiables. Los algoritmos más representativos dentro de este tipo de técnicas son: búsqueda tabú, optimización basada en colonia de hormigas, recocido simulado, entre otras.

Conceptos previos

Redes de datos

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 (Forouzan, 2007).

Existen diferentes tipos de clasificación de las redes de datos de acuerdo al contexto de estudio que se esté realizando. Según la topología, los esquemas más importantes son: árbol, bus, anillo, estrella o malla. De acuerdo al 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. En (Forouzan, 2007) se 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, como se muestra en la Figura 1.

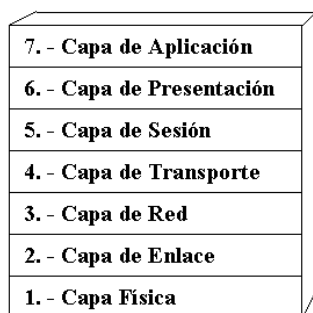


Figura 1. Modelo OSI.

Modelo TCP/IP

Es la definición de una arquitectura de protocolos de red. Su desarrollo se llevó a cabo en la década de 1970, 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, es por esto, que TCP/IP es considerada como su evolución. Está constituido por cinco capas independientes entre sí:

- Capa de red: Hace referencia a las definiciones físicas de la interface entre un dispositivo de transmisión de información y la red, como conexiones por aire a satélites, dispositivos de encaminamiento, cables y la tarjeta de interface de red.
- Capa de acceso a red: Permite el intercambio de información entre dos entidades conectadas a la red. Se especifican características como direccionamiento físico, topología de red, notificación de errores, secuencia de estructuras de datos y control de flujo.
- Capa de Internet: Brinda mecanismos para que la información pueda atravesar las diferentes redes que intervienen entre el emisor y el receptor. Este 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.
- Capa de transporte: La capa de transporte por medio de los protocolos TCP y UDP es la encargada de transmitir la información desde un proceso a otro de manera confiable, esto garantiza que la información llegue completa y en el orden en que fue enviada.
- El protocolo UDP (User Datagram Protocol) es un protocolo que por ser no orientado a conexión, no requiere un enlace lógico entre el emisor y el receptor, no garantiza la entrega de la información, ni el orden secuencia o protección frente a duplicados.

- El protocolo TCP (Transmission Control Protocol) permite la transmisión fiel de datos ya que es orientado a conexión. TCP divide un flujo de datos en unidades más pequeñas denominadas segmentos. Cada segmento incluye un número de secuencia para su reordenación en el receptor, junto con un número de confirmación para los segmentos recibidos.

- Capa de aplicación: Ofrece un amplio rango de protocolos, así como aspectos de codificación y control de diálogo. Algunos de los protocolos más importantes que ofrece este nivel son: DSN (Domain Name System) Sistema de nombre de dominio, FTP (File Transfer Protocol) Protocolo de transferencia de archivos, SMTP (Simple Mail Transfer Protocol) Protocolo simple de transferencia de correo.

Tipos de transmisión

Desde el punto de vista del transmisor y el receptor se puede clasificar como:

- Transmisión unidestino, que es realizada entre un emisor y un receptor, es decir, existe una relación uno a uno entre los dispositivos involucrados.
- Transmisión de difusión. Ejecutada en un único proceso desde un solo emisor a todas las máquinas conectadas a una red, no requiere realizar ningún tipo de copia en el origen para enviar el datagrama IP a varios destinatarios.
- Transmisión de multidifusión; definida como el tipo de transmisión entre un emisor y varios receptores, establece una relación de uno a muchos. Las unidades de datos o datagramas IP son enviados desde la máquina de 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 alcanzar la realización de 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.

- De esquema centralizado: Las decisiones de encaminamiento son tomadas por un único dispositivo 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 radica en que el núcleo, tiene toda la información referente a la red.

- De esquema distribuido: Actualmente, es utilizado en la mayoría de redes, 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

- Encaminamiento basado en el vector distancia: Los protocolos de este tipo son generalmente utilizados en redes

pequeñas. Su característica principal está en 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 el 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 cuestión 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), implican 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.

Una solución a un problema de optimización usualmente requiere un algoritmo configurable, el cual cuando se aplica a una instancia del problema, produce la solución deseada. Un algoritmo es un procedimiento finito paso a paso para solucionar un problema o para lograr un resultado requerido.

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. En estos casos se usan dos tipos de algoritmos.

- Algoritmos exactos. El algoritmo que busca solucionar el problema se ejecuta en un tiempo polinomial (Michiels, Aarts, y Korst, 2007). Generalmente se utiliza un algoritmo polinomial aunque no siempre es necesario. 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. Así, al crecer el tamaño del problema, el tiempo que tarda el algoritmo en resolverlo crece, pero despacio (polinómicamente), ya que la velocidad de crecimiento de un polinomio es lenta.

- Los algoritmos de aproximación, que generalmente se aplican 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. Dentro de este tipo de algoritmos se puede hacer una división, entre algoritmos heurísticos y metaheurísticos.

Algoritmos heurísticos. Los algoritmos heurísticos son los más fáciles de utilizar, ya que se basan en el conocimiento de una heurística que guía el proceso de búsqueda. El conocimiento del problema usualmente ayuda a encontrar una heurística razonable que encontrará rápidamente una solución aceptable. Un algoritmo de este tipo sólo 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) (Duarte, 2007). 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 (Duarte, 2007). Entre los algoritmos metaheurísticos más conocidos están, el recocido simulado 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 metalúrgico de recocido, que mediante un enfriamiento lento consigue un sólido más resistente (Rios y Rios, 2008).

Esta estrategia al principio 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; al 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, se ejecuta también muy lento. Lo que generalmente se

hace es mantener la temperatura constante durante L iteraciones, luego se reduce multiplicándose con α (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. (1) Parar si la función no ha mejorado en al menos ϵ_1 % tras k_1 iteraciones. (2) Parar si el número de transiciones aceptadas es menor que el ϵ_2 % de L tras las k_2 iteraciones (Rios y Rios, 2008).

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 memoria adaptativa y exploración sensible. La memoria adaptativa es el mecanismo por el que la búsqueda tabú asegura que las soluciones ya se encuentran en pasos anteriores para no repetirlos (Donoso y Fabregat, 2007). 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 a 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 función a 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 a 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 cual no se puede mejorar una parte (función) de la solución sin perjudicar alguna de las otras partes.

Actualmente, la optimización multiobjetivo es una de las vertientes más activas en el campo de las metaheurísticas, proponiendo versiones de los algoritmos metaheurísticos de objetivo simple, con algunos cambios conceptuales, sobre todo en la definición de la función objetivo que se debe optimizar. Así mismo, se plantean híbridos, que toman conceptos de inteligencia artificial, de economía, entre otros.

- 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 (Ecuación 1).

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

$$0 \leq r_i \leq 1 \quad / \quad i = 1, 2, 3, \dots, n \quad (1)$$

$$\sum_{i=1}^n r_i = 1$$

- 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 ϵ_k , de manera que cambiando los valores de ϵ_k para cada una de las nuevas restricciones se obtenga diferentes resultados de optimización (Ecuación 2).

$$f_k(X) \leq \epsilon_k, \quad k = 1, 2, 3, \dots, n \quad \text{y} \quad k \neq i \quad (2)$$

Propuesta de herramienta software

Análisis del problema

En esta etapa se establece el modelo a optimizar, especificando el dominio del problema combinatorio y la propuesta dentro del marco de los algoritmos metaheurísticos.

Modelamiento de la red. En una red de computadores (como por ejemplo una red local), el modelamiento de esta puede ser representado por medio de un grafo, en el cual los vértices pueden identificarse como los equipos de comunicaciones dispuestos para el tránsito de información (tales 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 a ser transmitidos por la red, se denotan

con F , y un flujo f pertenece a F . Así mismo, 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.

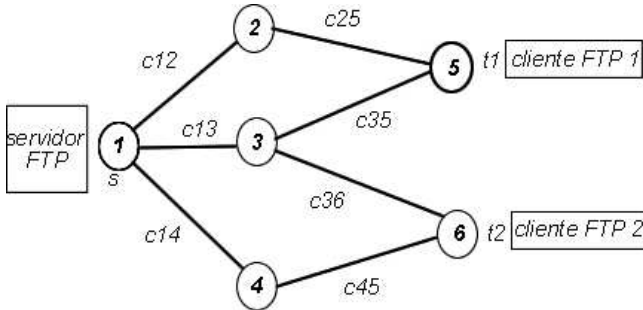


Figura 2. Transmisión de un flujo de datos FTP.

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

Tabla 1

Parámetros en transmisión TCP/IP

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

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}^{tf} , la cual se define como (Ecuación 3):

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

De esta manera, si es utilizada una arista dentro del flujo, se sumará un 1 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 1; las demás aristas que no son utilizadas tiene un 0.

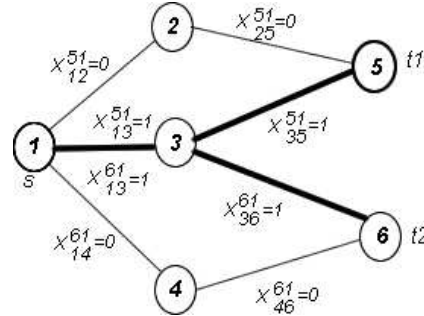


Figura 3. Transmisión de un flujo de datos con ruta utilizada.

Funciones a optimizar

El 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 (Ecuación 4):

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

Donde:

- $\sum_{(i,j) \in E} X_{ij}^{tf}$ 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.

El Retardo hace referencia a la detención 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 (Ecuación 5):

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

Donde:

- $\sum_{(i,j) \in E} d_{ij} \cdot X_{ij}^{tf}$ Es el retardo de propagación de las rutas para el flujo f que va para t .

El Costo está asociado con el valor monetario (la unidad de medida es en dinero) al 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 a transmitir y lo reenvía a los destinos indicados sin necesidad de duplicar la recepción del paquete. La función matemática asociada es la siguiente (Ecuación 6):

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

Donde:

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

El 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). 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 puede usarse 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 vez a través de dichos enlaces; esto es así porque el vértice (enrutador) hace una copia del paquete a transmitir y lo reenvía a los destinos indicados sin necesidad de duplicar la recepción del paquete. Esto permite que se optimice aun más la transmisión. La función matemática que expresa la función y muestra la evitación de envíos redundantes es (Ecuación 7):

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

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 (Ecuación 8):

$$c_1 = \sum_{(i,j) \in E} X_{ij}^{tf} = 1, t \in T_f, f \in F, i = s \quad (8)$$

- **Vértice destino:** Para cada flujo f , solamente una ruta debe llegar a cada vértice destino t . La expresión matemática es (Ecuación 9):

$$c_2 = \sum_{(i,j) \in E} X_{ij}^{tf} = -1, i = t, t \in T_f, f \in F \quad (9)$$

- **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 0 para todos los flujos. La expresión matemática es (Ecuación 10):

$$c_3 = \sum_{(i,j) \in E} X_{ij}^{tf} - \sum_{(j,i) \in E} X_{ji}^{tf} = 0, t \in T_f, f \in F, i \neq s, i \in T_f \quad (10)$$

- **Capacidad del enlace:** Evita que se envíe una demanda de tráfico más grande que la capacidad del enlace. La expresión matemática es (Ecuación 11):

$$c_4 = \sum_{f \in F} bw_f \cdot \max(X_{ij}^{tf})_{t \in T_f} \leq c_{ij}, (i, j) \in E \quad (11)$$

Donde:

- $\sum_{f \in F} bw_f \cdot \max(X_{ij}^{tf})_{t \in T_f}$ 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.

Metodología

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 prototipo de software contempla la optimización de las siguientes funciones objetivo:

- 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 (Ecuación 12):

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

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 hará utilizando las metaheurísticas trayectoriales de Recocido simulado y Búsqueda tabú, teniendo en cuenta algunos detalles.

La herramienta seleccionada 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 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 (Figura 4), 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 solicitudes de la vista, haciendo el llamado a 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, permitiendo que se pueda alterar alguno de ellos sin que eso implique cambios en la otra capa.

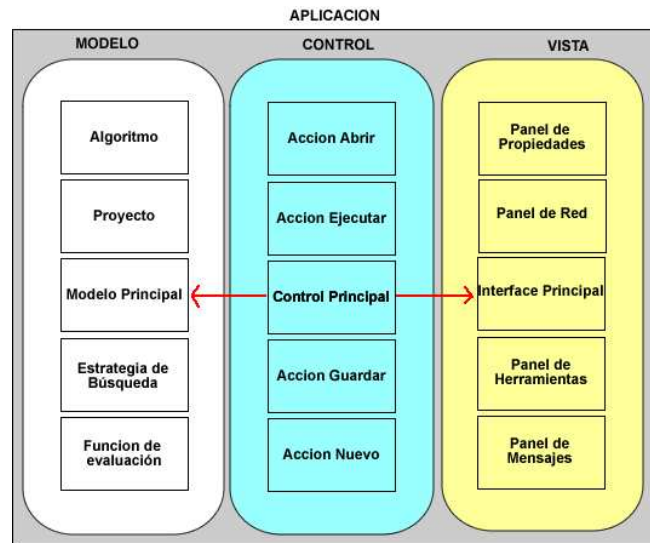


Figura 4. Modelo de diseño guiado por el patrón Modelo Vista Controlador.

Resultados

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 parámetros de los algoritmos a resolver y su posterior ejecución del proceso de optimización para obtener resultados comparables de la efectividad del algoritmo (Figura 5).

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 a través del gráfico de la red, donde 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) (Figura 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 (Figura 7).

La otra forma de ver los resultados, se hace a través de reportes multiobjetivo, en los que se muestran los valores de tres de las cuatro funciones objetivo, con la posibilidad de escoger cuáles visualizar (Figura 8).

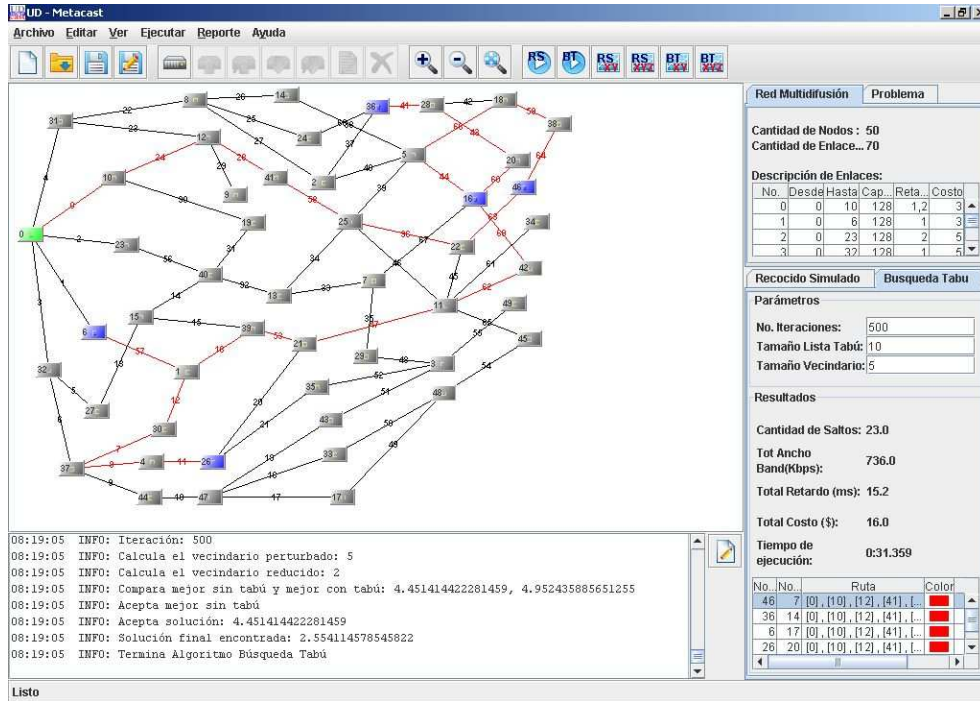


Figura 5. Interfaz UD-Metacast.



Figura 6. Ventana de algoritmos metaheurísticos.

Conclusiones y trabajo futuro

Se implementó el algoritmo Recocido Simulado, el cual utiliza el algoritmo Búsqueda en Profundidad como estrate-

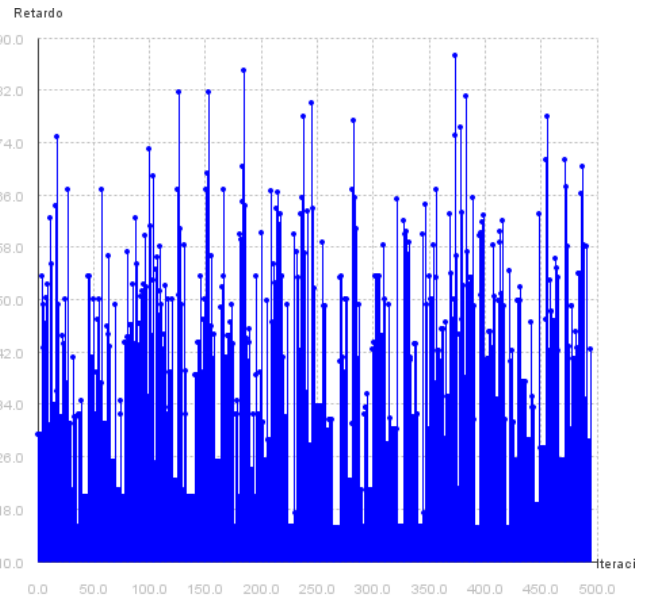


Figura 7. Reporte estadístico de la evolución del proceso de optimización.

gia de búsqueda local. Para reducir paulatinamente el valor de la temperatura se utilizó el esquema de enfriamiento de Cauchy. Como criterio de aceptación se escogió el algoritmo de Metrópolis el cual está basado en la distribución de probabilidad de Boltzmann. Para evaluar si una solución cumplía las restricciones de cada función objetivo se usó el esquema

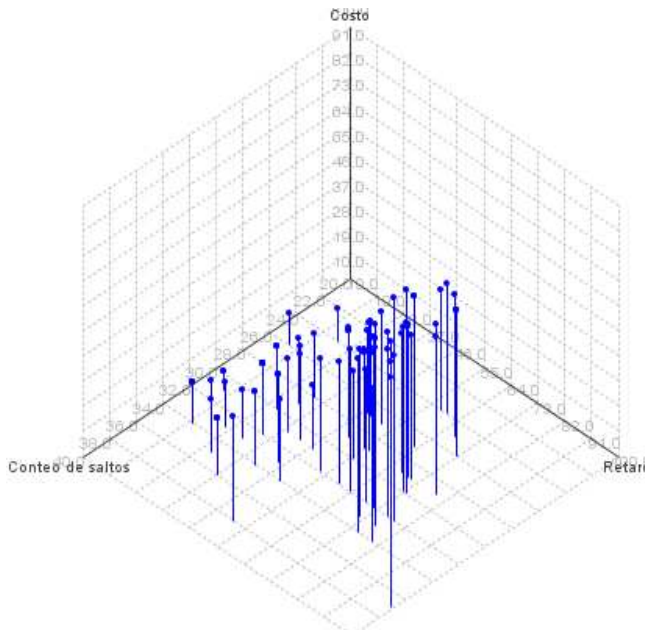


Figura 8. Reporte multiobjetivo del proceso de optimización.

de normalización lineal. Para la implementación de la Búsqueda Tabú, se usó memoria a corto plazo, mantenida con un tamaño, de manera que los movimientos que ingresen a la lista tabú, estarán allí durante la cantidad de iteraciones que indique el tamaño de la lista. Se estableció un tamaño para el vecindario a partir de una solución. Dicho tamaño será fijo y permitirá que se pueda tener grafos de grandes dimensiones sin que esto implique su exploración exhaustiva. La manera como las funciones objetivo son sopesadas en un solo valor, para que las metaheurísticas hagan uso de él en el proceso, fue por medio del método de suma ponderada de funciones. Dicha suma permite dar un grado de importancia a cada función, y devolverá un valor que representa el *pareto front* de la función. Por tratarse de funciones de diversos tipos de información, y por lo tanto de diferentes rangos, se hace un proceso de normalización de los datos antes de aplicar el método multiobjetivo.

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, donde el tamaño y las funcionalidades de las redes, no permiten que se pueda producir 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, y 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 resultó 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.

Como trabajo futuro, se plantean las siguientes recomendaciones.

- Implementar variaciones del algoritmo Recocido Simulado, cambiando aspectos como la estrategia de búsqueda local, el esquema de enfriamiento, el criterio de aceptación y/o la técnica de normalización; o adicionando nuevas características como el uso de memoria.

- Plantear 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.

- Adicionar nuevas funciones y características a la vista. Es posible permitir la ejecución de varios algoritmos a la vez o mantener el resultado de varias ejecuciones que permitan la comparación de resultados. Para el caso en el que se requiera trabajar con redes que tengan gran cantidad de nodos, es posible brindar la opción de construcción automática de la red, basada en esquemas o la adición de propiedades a los enlaces, utilizando valores aleatorios entre un mínimo y un máximo definido por el usuario.

- Adicionalmente, por tratarse de un proyecto relacionado con el área de investigación en inteligencia artificial, queda abierta la posibilidad de que se plantee el mismo problema (redes de multidifusión) en contextos más actuales, como los algoritmos evolutivos.

- Es posible también 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

- Donoso, Y., y Fabregat, R. (2007). *Multi-objective optimization in computer networks using metaheuristics* (1.^a ed.). Auerbach Publications.
- Duarte, A. (2007). *Metaheurísticas*. Dykinson.
- Forouzan, B. (2007). *Transmisión de datos y redes de comunicaciones* (4.^a ed.). McGraw Hill.
- Michiels, W., Aarts, E., y Korst, J. (2007). *Theoretical aspects of local search*. Springer.
- Rios, D., y Rios, S. (2008). *Simulación: Métodos y aplicaciones* (2.^a ed.). Ra-Ma.