

Un algoritmo de enrutamiento distribuido para redes de comunicación basado en sistemas de hormigas

J. Aguilar, *Member IEEE* y M. A. Labrador, *Senior Member IEEE*

Resumen-- En este artículo se propone un algoritmo de enrutamiento distribuido para redes de comunicaciones basado en Sistemas de Hormigas. El problema de enrutamiento para redes de comunicaciones es modelado como un problema de optimización combinatoria dinámico. En el algoritmo propuesto, el espacio de soluciones del problema de optimización combinatoria dinámico será el espacio donde pasean las hormigas, y la probabilidad de transición y la función de actualización de la tabla de feromona son definidos según la función objetivo del problema de comunicación. La naturaleza general del enfoque propuesto permite su utilización en diferentes redes de comunicación, cambiando únicamente el criterio de rendimiento a optimizar. Los resultados obtenidos después de comparar el algoritmo propuesto en diferentes ambientes de comunicación demuestran que esta propuesta provee mejores rendimientos en cuanto al retraso y eficacia en el uso de energía, entre otras medidas.

Palabras clave-- Algoritmos de enrutamiento, optimización combinatoria dinámica, procesamiento inteligente distribuido, redes inalámbricas, redes sensoriales, redes tolerantes a fallos.

I. INTRODUCCIÓN

EL problema a resolver por cualquier sistema de enrutamiento es dirigir el tráfico de las fuentes a los destinos mientras se maximiza algún criterio de rendimiento de interés. Dependiendo del tipo de red, los criterios de rendimiento pueden ser la proporción de rechazo de llamadas, la tasa de transmisión, el retraso o la energía, entre los más importantes. El enrutamiento en las redes de comunicación es necesario porque en los sistemas reales no todos los nodos se conectan directamente. Actualmente, los algoritmos de enrutamiento enfrentan desafíos importantes debido a la complejidad encontrada en las redes modernas. Por ejemplo, los algoritmos centralizados tienen problemas de escalabilidad, los algoritmos estáticos no pueden mantener al ritmo de los cambios constantes de las redes, y los algoritmos distribuidos y dinámicos tienen problemas de estabilidad y oscilación.

En general, los algoritmos de enrutamiento deben enfrentar problemas en las redes modernas, tales como las condiciones de tráfico, la estructura de la red, y los recursos de la red (los cuales son limitados y están cambiando constantemente). Esto es particularmente cierto en las redes inalámbricas y sensoriales, donde la movilidad de los nodos y los fallos de los dispositivos producen cambios constantes en la topología de la red. Para estas redes, los algoritmos de enrutamiento dinámicos son el único enfoque factible. De hecho, la falta de adaptabilidad de los algoritmos de enrutamiento para adaptarse a los frecuentes cambios de la topología de la red, de las capacidades de los nodos, de los modelos de tráfico, de la carga, de la disponibilidad de energía, entre otros, reduce el rendimiento en las redes. Este problema puede definirse como un problema de optimización combinatoria dinámico distribuido [2, 10], un desafío en el dominio de optimización combinatoria.

Los denominados Sistemas Artificiales de Hormigas (SAH)[3] proporcionan una alternativa prometedora para desarrollar algoritmos de enrutamiento para las redes de comunicación modernas. Agentes de software móviles y autónomos tienen la capacidad para adaptarse, cooperar y moverse inteligentemente de un sitio a otro en la red de comunicación. Las propiedades inherentes de los SAH incluyen la escalabilidad, la posibilidad de descubrir nuevas vías y soluciones, y la inteligencia, la baja complejidad de las interacciones locales y la comunicación a través del sistema, que son los rasgos deseables para muchos tipos de redes. En general, las hormigas reales son capaces de hallar el camino más corto entre una fuente de comida y su nido segregando sustancias químicas llamadas feromonas. Las hormigas hacen uso de feromonas marcadoras de pistas para indicar el mejor camino a las siguientes [1, 3, 4, 5, 7]. Este comportamiento de las hormigas reales ha inspirado los SAH, algoritmos en los que un grupo de hormigas artificiales transitan por la red de comunicaciones y cooperan para la solución de un problema intercambiando información mediante la feromona depositada en el grafo.

Los SAH se han usado en el pasado para resolver otros problemas de optimización combinatoria, como el problema del viajero de comercio y el problema de la asignación cuadrática, entre otros [3, 4, 5, 6, 7, 8]. Nosotros hemos propuesto un algoritmo distribuido basado en conceptos de los SAH llamado Sistemas de Hormigas Combinatorias (SHC), para resolver problemas de optimización combinatoria estáticos y dinámicos [1, 2]. La idea principal introducida por

Este trabajo fue financiado en parte por el CDCHT-ULA (Proyecto I-820-05-02-AA), y el Fonacit (Proyecto 2005000170).

Jose Aguilar trabaja en CEMISID. Departamento de Computación, Facultad de Ingeniería, Universidad de los Andes, Mérida, 5101, Venezuela (correo e.: aguilar@ula.edu).

Miguel A. Labrador trabaja en el Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA (correo e.: labrador@cse.usf.edu).

nuestro modelo es la definición de un procedimiento general para resolver problemas de optimización combinatoria usando SAH. En nuestro enfoque, el grafo que describe el espacio de soluciones del problema de optimización combinatoria es el grafo del SAH, la función de transición y la fórmula de actualización de la tabla de feromona. Esta tabla, la cual se construye según la función objetivo del problema de optimización combinatoria, guarda la cantidad de feromona asignada a cada una de las distintas posibilidades o caminos a seguir. En este artículo, presentamos un algoritmo de enrutamiento basado en SHC porque este problema puede definirse como un problema de optimización combinatoria dinámico.

La generalidad del algoritmo de enrutamiento propuesto se evalúa en tres tipos de redes de comunicación. Primero, el algoritmo se aplica a redes estáticas, es decir las redes sin cambios en la topología. Después, estudiamos redes con frecuentes cambios en la topología debido a fallos imprevisibles en los nodos o arcos de interconexión, como en las redes inalámbricas del tipo ad hoc. Finalmente, el algoritmo propuesto también se aplica a las redes con restricciones de energía, como las redes inalámbricas sensoriales donde los nodos son muy pequeños y baratos, y con una energía y capacidades de cómputo y comunicación limitados [9, 12, 17].

El resto del artículo está organizado de la siguiente manera. La sección 2 presenta el SHC y el problema de enrutamiento. La sección 3 presenta el algoritmo de enrutamiento distribuido general basado en el SHC. Después, la sección 4 muestra la utilización de este algoritmo en tres redes de comunicación diferentes. Finalmente, se presentan las conclusiones en la sección 5.

II. ASPECTOS TEÓRICOS

A. Sistemas Combinatorios basados en Hormigas

Inteligencia colectiva aparece en diferentes sistemas sociales, como los sistemas biológicos de ciertas especies de insectos. Da lugar a un complejo comportamiento inteligente a través de la interacción de miles de miembros autónomos del sistema. El resultado es lograr comportamientos sociales muy complejos de optimización de tareas [1, 3, 4, 5, 8]. El principio fundamental detrás de estas interacciones es la reacción autocatalítica, como en el caso de los Sistemas de Hormigas, donde las hormigas atraídas por un conjunto de feromonas siguen ese sendero, causando que nuevas hormigas sean también atraídas.

Las hormigas tienen una gran capacidad de supervivencia que se deriva de un proceso de Conducta Colectiva [3] y se basa en sus capacidades de comunicación. La comunicación entre agentes (las hormigas) se realiza mediante un rastro químico denominado feromona. De esta forma, una hormiga deja una cierta cantidad de feromona en el sendero cuando se mueve. La probabilidad que una hormiga siga un camino/sendero depende del número de hormigas que lo hayan tomado antes (una cantidad grande de feromona en un camino significa una mayor probabilidad de que lo sigan otras hormigas).

EL SAH es el progenitor de todos los esfuerzos en investigación con algoritmos de hormigas, y se aplicó primero al Problema del Viajero de Comercio (VC) [5, 6, 7]. Algoritmos inspirados por ellos se han mostrado como métodos heurísticos que resuelven problemas de optimización combinatoria. Estos algoritmos se caracterizan por su versatilidad, robustez y funcionamiento basado en poblaciones. El procedimiento se basa en la búsqueda por agentes, llamados "hormigas", que tienen capacidades muy simples y que intentan simular la conducta de las hormigas.

El SAH utiliza una representación de grafo (grafo AS), donde cada arco (r,s) tiene una medida de conveniencia γ_{rs} , llamado feromona, que es actualizada por las hormigas artificiales cuando pasan por él. Informalmente, el procedimiento siguiente ilustra cómo trabaja el SHA. Cada hormiga genera un viaje de ida y vuelta escogiendo los nodos según una regla de transición; las hormigas prefieren moverse a nodos que se conectan por arcos que tienen una presencia de feromona alta. Una vez todas las hormigas han completado sus viajes se realiza una actualización global de los rastros o feromona. La actualización consiste en que cada hormiga deposita una cantidad de feromona en los arcos correspondiente a la longitud del recorrido que ella hizo y, además, una parte de la feromona existente se evapora de todos los arcos.

Hay dos razones para usar el SAH en el VC. Primera, el grafo de VC puede trazarse directamente sobre el grafo AS. Segunda, la función de la transición del SAH tiene las metas similares a la función objetivo a optimizar en el problema del VC. Este no es el caso para otros problemas de optimización combinatoria. En [1, 2] nosotros propusimos un algoritmo distribuido basado en conceptos del SAH, llamado SHC, que resuelve cualquier tipo de problema de optimización combinatoria. En este caso, cada hormiga construye una solución al caminar a través del grafo AS, usando una regla de transición y un rastro de feromonas que es actualizado por una fórmula definida según la función objetivo del problema de optimización combinatoria. Este proceso involucra los pasos siguientes:

1. Definir el grafo que describe el espacio de soluciones del problema de optimización combinatoria. Así, el espacio solución se define mediante un grafo donde los nodos representan las posibles soluciones parciales al problema, y los arcos la relación entre las soluciones parciales.
2. Construir el grafo AS. Para esto se usa el grafo construido en la fase 1, este nuevo grafo será en el que las hormigas recorrerán.
3. Definir la función de transición y la fórmula para actualizar la tabla de feromona del SHC. Estos se construyen según la función objetivo del problema de optimización combinatoria.
4. Ejecutar el procedimiento del SAH descrito antes.

1) Definir el grafo que describe el espacio de la solución del problema de optimización combinatoria.

El primer paso es construir el grafo del problema de optimización combinatoria, para después definir el grafo AS con la misma estructura que el grafo del problema de

optimización combinatoria. El grafo AS tiene dos matrices de peso. La primera matriz se define según el grafo del problema de optimización combinatoria y registra las relaciones entre los elementos del espacio de solución (matriz del problema de optimización combinatoria). La segunda registra la feromona dejada en cada arco (la matriz de feromona). Esta matriz de pesos es actualizada según la fórmula de actualización de feromona. Cuando los pesos de un arco de la matriz de feromona para un nodo dado son altos, este nodo tiene una probabilidad alta de ser visitado. Por otro lado, si un arco entre dos nodos de la matriz del problema de optimización combinatoria es bajo, entonces significa que si uno de estos nodos pertenece a la solución entonces el otro también debe pertenecer. Si el arco es igual al infinito entonces significa que los nodos son incompatibles, y por consiguiente, ellos no pueden pertenecer a la misma solución final. Nosotros definimos una estructura de datos para guardar la solución que cada hormiga k está construyendo. Esta estructura de datos es un vector (A_k) con una longitud igual a la longitud de la solución, y determina el número de nodos que una hormiga debe visitar. Para una hormiga dada, el vector guarda cada nodo del grafo SAH que visita.

2) Definir la función de transición y la fórmula para actualizar la tabla de feromona del SHC.

La regla de transición y la fórmula de actualización de feromona se construyen usando la función objetivo del problema de optimización combinatoria. La función de la transición entre nodos es:

$$Tf(\gamma_{rs}(t), Cf_{r \rightarrow s}^k(z)) = \frac{\gamma_{rs}(t)^\alpha}{Cf_{r \rightarrow s}^k(z)^\beta}$$

donde $\gamma_{rs}(t)$ es la feromona en la iteración t , $Cf_{r \rightarrow s}^k(z)$ es el costo de la solución parcial que está construyendo la hormiga k cuando cruza el arco (r, s) si está en la posición r , $z - l$ es la longitud actual de la solución parcial, y, α y β son dos parámetros ajustables que controlan el peso relativo de la intensidad de la traza $(rs(t))$ y la función de costo.

En el SHC, la probabilidad de transición es como sigue: una hormiga k que esta en el nodo r escoge el nodo s para moverse según una probabilidad $P_{rs}^k(t)$ que es calculada según la ecuación 1:

$$P_{rs}^k(t) = \begin{cases} \frac{Tf(\gamma_{rs}(t), Cf_{r \rightarrow s}^k(z))}{\sum_{u \in J_r^k} Tf(\gamma_{ru}(t), Cf_{r \rightarrow u}^k(z))} & \text{Si } s \in J_r^k \\ 0 & \text{de lo contrario} \end{cases} \quad (1)$$

donde J_r^k es el conjunto de nodos conectado a r que aun no han sido visitados por la hormiga k posicionada en el nodo r . Cuando $\beta=0$ solo se usan las soluciones anteriores (sólo la intensidad de la traza se usa), y cuando $\alpha=0$ se explora el espacio de soluciones (es un algoritmo estocástico de búsqueda). Un intercambio entre la calidad de soluciones parciales e intensidad de la traza es necesario. Una vez todas

las hormigas han construido sus soluciones, la feromona es actualizada (es decir, la intensidad de la traza) en la matriz de feromona, en todos los arcos según la ecuación 2 [1, 2, 3, 4, 5, 6, 7]:

$$\gamma_{rs}(t) = (1 - \rho)\gamma_{rs}(t-1) + \sum_{k=1}^m \Delta\gamma_{rs}^k(t) \quad (2)$$

donde ρ es un coeficiente tal que $(1-\rho)$ representa la evaporación de la traza en una iteración (un giro en el cual cada hormiga ha propuesto una solución), m es el número de hormigas, y $\Delta\gamma_{rs}^k(t)$ es la cantidad por unidad de longitud de feromona puesta en el arco (r,s) por la hormiga k en esa iteración:

$$\Delta\gamma_{rs}^k(t) = \begin{cases} \frac{1}{C_f^k(t)} & \text{Si arco } (r,s) \text{ ha sido cruzado por la hormiga } k \\ 0 & \text{de lo contrario} \end{cases} \quad (3)$$

donde $C_f^k(t)$ es el valor de la función de costo (función objetivo) de la solución propuesta por la hormiga k en la iteración t . El procedimiento general es como sigue:

1. Generación de la grafica AS.
2. Definición de la regla de transición y la formula para actualizar la feromona, de acuerdo con el problema de optimización combinatorio.
3. Repetir hasta que el sistema alcanza una solución estable.
 - 3.1. Poner m hormigas en diferentes nodos de la grafica AS.
 - 3.2. De $i=1, n$
 - 3.2.1. De $j=1, m$
 - 3.2.1.1. Escoger nodo s de acuerdo a la probabilidad de transición (Ecuación 1).
 - 3.2.1.2. Mover la hormiga m al nodo s .
 - 3.3. Actualizar la tabla de feromona utilizando las formulas de actualización de feromona (Ecuaciones 2 y 3).

B. Problema de Enrutamiento

Enrutamiento es la tarea que permite transmitir la información en una red desde una fuente a un destino a través de una sucesión de nodos intermedios. El enrutamiento es necesario porque en los sistemas reales no todos los nodos se conectan directamente. Los algoritmos de enrutamiento pueden ser clasificados como estáticos o dinámicos, y

centralizados o distributivos [11, 13, 16]. Los algoritmos centralizados normalmente tienen los problemas de escalabilidad, y la incapacidad de la red para recuperarse en caso de una falla de la estación central. El enrutamiento estático asume que las condiciones de la red son invariantes en el tiempo, que es una asunción poco realista en la mayoría de los casos. Los esquemas de enrutamiento adaptables también tienen problemas, incluso inconsistencias cuando hay fallos en los nodos o muchas oscilaciones en la carga de trabajo. Los algoritmos de enrutamiento también pueden ser clasificados como mínimos o no mínimos [11, 13, 16]. El enrutamiento mínimo permite que los paquetes sólo sigan caminos con el costo mínimo, mientras que el enrutamiento no mínimo tiene más flexibilidad escogiendo el camino utilizando otras heurísticas. Otra clase de algoritmo de enrutamiento es uno donde el esquema de enrutamiento debe garantizar especificaciones de Calidad de Servicio [11, 16].

Normalmente, las redes modernas utilizan esquemas de enrutamiento dinámicos para cubrir los cambios constantes en las condiciones de tráfico y en la estructura o topología de la red. Éste es particularmente el caso de las redes inalámbricas donde la movilidad de los nodos y los fallos son imprevisibles y frecuentes, lo que producen cambios en la topología de la red. Una inmensa literatura de algoritmos de enrutamiento existe especialmente para estos tipos de redes [20, 31, 32, 33, 34, 36], todos ellos con la meta principal de hacer la red más confiable y tolerante a fallos [10]. Sin embargo, aumentar al máximo el servicio de la red para condiciones de carga y topología de la red variantes en el tiempo es un problema NP-completo. Un algoritmo de enrutamiento con estas características puede definirse como un problema de optimización combinatoria dinámico, es decir como un problema distribuido variante en el tiempo. En este trabajo, nosotros vamos a usar a nuestro modelo *SHC* para proponer un algoritmo de enrutamiento para estas redes, que incluso pueden tener nodos que fallan.

Las redes sensoriales también presentan el problema de enrutamiento como algo desafiante. Típicamente, estas redes consisten en miles, o incluso ciento de miles de nodos, desplegados en un área arriesgada o inaccesible para supervisar y controlar [12, 13, 17, 26]. Al contrario de las otras redes inalámbricas, los nodos son bastante estáticos y periódicamente transmiten muchos datos de proporciones muy bajas. Además, los nodos son muy limitados en lo que se refiere a recursos, energía, y capacidades de procesamiento y comunicación. Es conocido que de todos estos factores la conservación de energía es la más importante, ya que se relaciona directamente con la vida de la red, por lo cual el esquema de enrutamiento debe ser sensible a eso. El problema actual en los protocolos actuales para las redes sensoriales es que ellas encuentran la ruta de energía más baja y la usan para cada comunicación. Sin embargo, eso no es lo mejor para extender la vida de la red. Frecuentemente, al usar el camino de energía más bajo lleva al vaciamiento de energía a los nodos a lo largo de ese camino más rápido, conllevando a una partición de la red. Para aumentar la supervivencia de la red, un protocolo de enrutamiento se propone en [13] que usa los caminos subalterno-óptimos de vez en cuando. Esto asegura

que el camino óptimo no se anula y la red se degrada menos. Nosotros vamos a usar nuestro modelo para proponer un mecanismo de enrutamiento para estas redes. El protocolo deberá ser escalable a un número grande de nodos, flexible a fallos de los nodos, y con uso eficaz de energía.

III. ALGORITMO DE ENRUTAMIENTO DISTRIBUIDO BASADO EN EL MODELO DE HORMIGAS COMBINATORIAS

En general, existe un importante número de investigaciones y proyectos basados en las capacidades de auto-organización de los sistemas de Hormigas, en telecomunicaciones, tanto para problemas de enrutamiento como para equilibrar la carga del tráfico. Por ejemplo, hay varios algoritmos de enrutamiento basados en hormigas propuestos en [3, 10, 11, 27, 28, 29, 30]. El más famoso es AntNet [27], un algoritmo de enrutamiento basado en agentes adaptables que tiene el mismo rendimiento que los mejores algoritmos de enrutamiento conocidos para redes de computadoras [3, 11]. También se han aplicado los sistemas de hormigas en redes de telefonía. El algoritmo ABC es otro ejemplo de aplicación exitosa de estos sistemas [3]. En este artículo proponemos un nuevo algoritmo de enrutamiento basado en nuestro modelo *SHC* que puede usarse en la gestión de diferentes redes, como las redes con topología estáticas, redes con cambios constantes en la topología, o redes de computadoras con restricciones de energía.

Nosotros podemos usar nuestro enfoque para conexiones punto a punto o punto-multipunto. En el caso punto a punto, una hormiga se usa para buscar el mejor camino al destino. Para una demanda multipunto con m destinos, se usan m hormigas y se selecciona la ruta con nodos intermedios con mayor cantidad de feromona. Para esto, usamos las tablas de enrutamiento locales de cada nodo como una función de transición a sus vecinos. Así, según el destino del mensaje, el nodo con la probabilidad más alta para ser visitado corresponde a la entrada en la tabla con la mayor cantidad de feromona. Entonces, la tabla de enrutamiento local se actualiza conforme a la ruta seleccionada. Nuestro algoritmo puede trabajar en redes estables (redes donde los cambios son suficientemente lentos para que las actualizaciones de enrutamiento se propaguen a todos los nodos) o no, porque nuestro enfoque sólo trabaja con las tablas de enrutamiento locales y los cambios sólo deben propagarse a los vecinos.

A. Construyendo el grafo AS

Nosotros usamos la matriz de feromona del grafo AS como la tabla de enrutamiento de cada nodo de la red. Recuerde que esta matriz es donde se deposita el rastro de feromona. Particularmente, cada nodo i tiene k_i vecinos, se caracteriza por una capacidad C_i , una capacidad de repuesta S_i , y por una tabla de enrutamiento $R_i = [r_{n,d}^i(t)]_{k_i, N-1}$. Cada fila de la tabla de enrutamiento corresponde a un nodo del vecino y cada columna a un nodo del destino. La información de cada fila de nodo i se guarda en el lugar respectivo de la matriz de feromona (por ejemplo, en la posición i, j si el vecino de $k_i = j$). El valor $r_{n,d}^i(t)$ se usa como una probabilidad. Es decir, la probabilidad que una hormiga dada, donde el destino es el

nodo d , se enrute del nodo i al nodo vecino n . Nosotros usamos la matriz de optimización combinatoria de nuestro grafo AS para describir la estructura de la red. Si hay fallos de nodos, entonces el grafo de optimización combinatoria se modifica para mostrar eso. Además, en cada arco del grafo de optimización combinatoria, la estimación del tiempo del viaje del nodo actual i a su nodo vecino j , es denotado $\Gamma_i = \{\mu_{i \rightarrow j}, \sigma_{i \rightarrow j}^2\}$, dónde $\mu_{i \rightarrow j}$ es el promedio del tiempo de viaje estimado del nodo i al nodo j , y $\sigma_{i \rightarrow j}^2$ es su variación asociada. Γ_i proporciona una idea local del estado de la red global al nodo i . Finalmente, nosotros definimos una función de costo para cada nodo, llamado $C_{ij}(t)$, éste es el costo asociado con este enlace. Es una variable dinámica que depende de la carga del enlace, y es calculado en el momento t usando Γ_i .

B. Definiendo la función de la transición y la fórmula de actualización de la tabla de feromona

Nosotros hemos explicado en nuestro modelo descentralizado que cada nodo mantiene una tabla de enrutamiento que indica donde el mensaje debe ir para alcanzar el último destino. Las hormigas artificiales ajustan las entradas de la tabla para mostrar el estado actual de la red continuamente. Así, las tablas de enrutamiento se comportan como una tabla de feromona que tiene la probabilidad de que las hormigas artificiales sigan un determinado camino. En nuestra red, cada hormiga actualiza sus influencias en la tabla del feromona aumentando o reduciendo la entrada para el destino apropiado.

En nuestro modelo, cada nodo de la red se representa como una estructura que contiene varios parámetros (la identificación del nodo, nodos adyacentes, la capacidad de respuesta, el número de enlaces), y la Ecuación 3 tiene el significado siguiente: $C_f^k(t)$ es el costo de la ruta de la k^{ma} hormiga, $\Delta\gamma_{is}^k(t)$ es la cantidad de feromona depositada por la k^{ma} hormiga si el arco (i, s) pertenece a la ruta de la hormiga k^{ma} (se usa para actualizar la tabla de enrutamiento Ri en cada nodo), y $P_{ij}^k(t)$ es la probabilidad que la hormiga k^{ma} escoja brincar del nodo i al nodo j (es calculado de la tabla de enrutamiento Ri). De esta manera, las hormigas caminan según las probabilidades en las tablas de feromona y ellos visitan un nodo cada vez. Hormiga k^{ma} actualiza su costo de ruta cada vez que cruza un enlace usando $C_f^k(t) = C_f^k(t) + C_{ij}(t)$, si el enlace (i, j) pertenece al camino seguido por la hormiga k^{ma} . De esta manera, una hormiga colecciona las colas y carga de tráfico experimentadas que le permiten definir la información sobre el estado de la red. Una vez ha alcanzado el nodo destino d , la hormiga k^{ma} regresa a su nodo fuente a través de todos los nodos visitados durante el camino, y actualiza las tablas de enrutamiento (la concentración de feromona) y las estimaciones de los tiempos de viaje entre los nodos que pertenecen a su camino (el grafo de optimización combinatoria), como sigue:

- El tiempo del camino $i \rightarrow d$ ($T_{i \rightarrow d}$) de la ruta de la hormiga k^{ma} actual se usa para actualizar los valores de las medias y varianzas Γ_i de los nodos i que

pertenecen a la ruta. $T_{i \rightarrow d}$ da una idea sobre la bondad de la ruta seguida porque su longitud es proporcional al tráfico o congestión.

- La tabla de enrutamiento Ri se cambia incrementando la probabilidad $r_{j,d}^i(t)$ asociado con el nodo vecino j que pertenece a la ruta de la hormiga k^{ma} , y disminuyendo la probabilidades $r_{j,d}^i(t)$ asociado con los otros n nodos vecinos, todos teniendo como destino a d , dónde $n \neq j$ para el mismo destino (es como la traza de feromona).

Se usan los valores guardados en Γ_i para cronometrar el tiempo de viaje, tal que se puedan transformar en una señal de refuerzo $r = 1/\mu_{i \rightarrow j}$, $r \in [0, 1]$. r es usado por el nodo actual i como un refuerzo positivo para el nodo j :

$$r_{i-1,d}^j(t+1) = r_{i-1,d}^j(t) (1-r) + r$$

y la probabilidad $r_{n,d}^i(t)$ de los otros n nodos vecinos cuyos destinos son d , reciben un refuerzo negativo:

$$r_{n,d}^i(t+1) = r_{n,d}^i(t) (1-r) \quad \text{para } n \neq j$$

De esta manera, las hormigas artificiales pueden seguir caminos que evitan la congestión mientras equilibran la carga de la red. Finalmente, $C_{ij}(t)$ se actualiza usando Γ_i y considerando el problema de congestión (se deben evitar los nodos congestionados):

$$C_{ij}(t+1) = C e^{-ds_j(t)} \frac{\mu_{i \rightarrow j}}{\sigma_{i \rightarrow j}^2} \quad (4)$$

donde C y d son constantes, y $s_j(t)$ es la capacidad de respuesta del nodo j en el momento t . La incorporación del retraso ($C e^{-ds_j(t)}$) reduce la proporción de flujo de las hormigas a los nodos congestionados. En el caso de fallos en enlaces, el algoritmo evita esos nodos según la fórmula siguiente:

$$C_{ij}(t+1) = \infty \quad (\text{nodo } j \text{ con falla}) \quad (5)$$

En el caso de una red sensorial, las hormigas buscan caminos que aumentan al máximo la utilización de la energía en cada nodo, usando la fórmula siguiente:

$$C_{ij}(t+1) = C e^{-dg(t)} \quad (6)$$

donde, C y d son constantes, y $g(t)$ es la vida de la batería en el momento t (ésta es una retroalimentación negativa). No obstante, nuestro costo puede incluir información sobre el costo de usar el camino, energía disponible en los nodos a lo largo del camino, topología de la red, etc. Por esta razón, la fórmula siguiente también puede usarse en las redes sensoriales [12]:

$$C_{ij}(t+1) = E_{ij}^\alpha R_j^\beta \quad (7)$$

donde E_{ij} es la energía usada para transmitir y recibir en el enlace, y R_j es la energía residual en el nodo j normalizado por la energía inicial del nodo. Los factores α y β pueden escogerse para encontrar el camino de energía mínimo, o el camino con nodos que tienen la mayor cantidad de energía, o una combinación de lo anterior.

IV. LA EVALUACIÓN DEL ALGORITMO PROPUESTO

En esta sección nosotros probamos nuestro enfoque considerando tres casos, redes con topología estática (ningún fallo), redes con cambios constantes en la topología debido a fallos en los nodos y enlaces, y redes con restricciones de energía con y sin fallos.

A. Redes con topología estática

Nosotros empleamos dos redes muy conocidas para evaluar nuestros algoritmos en redes con topología estables. La primera red es la NSFNET-T1 americana que consiste de 14 nodos y 21 enlaces bidireccionales con un ancho de banda de 1,5 Mbps y retrasos de la propagación que van de 4 a 20 ms. La segunda red es la NNTnet japonesa que tiene 57 nodos y 162 enlaces bidireccionales de 6 Mbps y la propagación es de 1 a 5 ms [2, 3, 15, 16]. Se han considerado varios modelos de tráfico diferentes, considerando características espaciales y temporales. Nosotros consideramos los modelos de tráfico Poisson (P) y Fijo (F) para la distribución temporal, y Uniforme (U) y al Azar (R) para la distribución espacial. Los modelos de tráfico generales son obtenidos entonces combinando las características anteriores. Así, estos modelos determinan el valor de m (número de hormigas), tal que por cada mensaje a transmitir tendremos una hormiga. El rendimiento de la red se expresa en la tasa de transmisión y en el tiempo de la fuente al destino. Los nodos tienen una capacidad de 40 llamadas. Así, cada llamada que usa un nodo disminuye la capacidad de respuesta del nodo en 2,5%. Para este problema, los valores con los que se obtuvieron los mejores resultados para las variables α y β de la ecuación 1 son 0,5 y 0,5, respectivamente, y para ρ de la ecuación 2 es 0,3. Nosotros comparamos nuestro algoritmo, primero con AntNET y el algoritmo SPF, el actual algoritmo de enrutamiento oficial de Internet [3, 11]. Las Figuras 1 y 2 muestran la tasa de transmisión y el resultado del retraso para una distribución de Poisson temporal y una distribución Aleatoria espacial del tráfico en NSFNET y NNTnet que usan la Ecuación 4. La Figura 3 es para una distribución Uniforme temporal y Fija espacial del tráfico. Estos resultados son ejemplos de la conducta de los algoritmos; resultados obtenidos utilizando otros modelos de tráfico y topologías de red son cualitativamente equivalentes y pueden encontrarse en [15].

La tasa de transmisión lograda por nuestro enfoque es por lo menos tan buena como el de AntNET, y los retrasos de paquete son mucho mejor que el de los otros. Al principio, nuestro enfoque no tiene el mejor rendimiento porque tiene que aprender la situación actual de la red, sin embargo, después encuentra la ruta óptima de una manera impresionante, proporcionando un buen rendimiento.

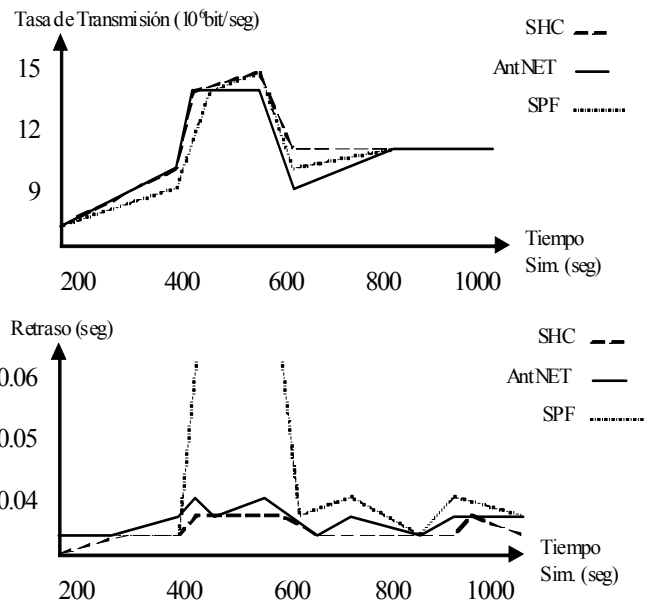


Figura 1. Comparación de los algoritmos en NSFNET con un tiempo entre llegadas medio de 2,4 segundos.

Igualmente, la Figura 2 muestra que nuestro algoritmo continúa dando el mejor rendimiento. En este caso, el retraso es más grande porque el número de enlaces en esa red es más grande.

La Figura 3 muestra los resultados que utilizan la combinación de tráfico Fijo y Uniforme. Como puede verse, nuestro algoritmo reduce el retraso de paquetes permanentemente y todavía proporciona una tasa de transmisión global mejor. Nuestro enfoque considera muy bien el problema de congestión y provee mejor rendimiento asignando quizás las rutas más largas, pero menos congestionadas.

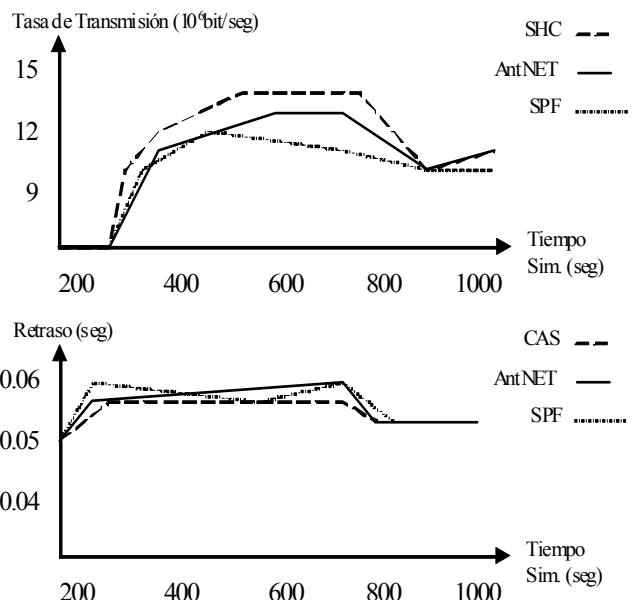


Figura 2. Comparación de los algoritmos en NNTnet usando un tiempo entre llegadas medio de 3,1 segundos.

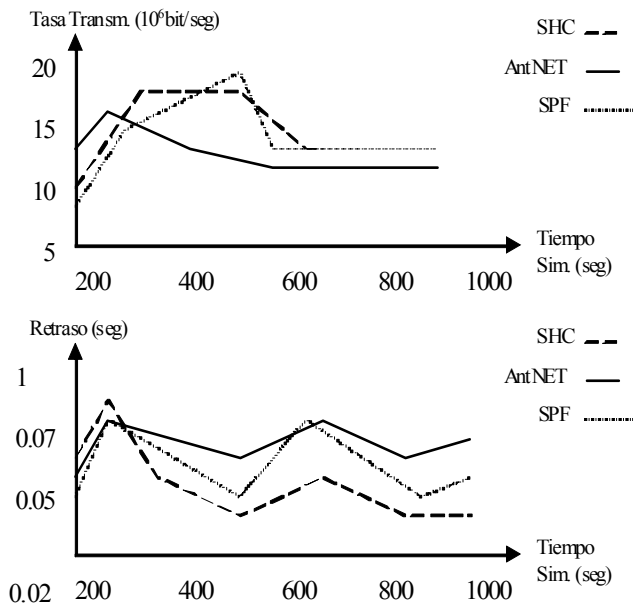


Figura 3. Comparación de los algoritmos en NSFNET con un tiempo entre llegadas medio de 3,1 segundos.

B. Redes con cambios constantes en la topología

En este experimento, nosotros evaluamos nuestro algoritmo en redes con cambios constantes de la topología debido a fallos en los enlaces y nodos. Aquí, si una falla en el enlace ocurre y el nodo tiene más de una unión, entonces el nodo puede alcanzar el nodo destino por otro camino. Si el nodo no tiene ningún otro enlace a cualquier nodo en la red, entonces un fallo de nodo ocurre. Nosotros asumimos que esa falla del enlace sigue una distribución uniforme y no excede el 10% del número total de enlaces en la red. En la presencia de una falla de enlace, el costo de una llamada del nodo fuente i al nodo destino j se definirá como infinito (vea Ecuación 5), y la probabilidad en la columna y fila apropiada en la tabla de feromona se pone en cero.

Como en [10], nosotros también consideramos la incorporación de ruido que modela los problemas de bloqueo y cortocircuito. El problema de cortocircuito ocurre cuando una ruta más corta se pone repentinamente disponible, mientras el problema del bloqueo ocurre cuando una ruta más vieja se pone no disponible. En ambas situaciones, las hormigas artificiales tienen dificultades para encontrar las nuevas rutas, ya que ellas trabajan guiadas por las tablas de feromona y no tienen una reacción dinámica adecuada. Con la inclusión del factor de ruido f , las hormigas seleccionan un camino completamente aleatorio con probabilidad f y un camino guiado por la tabla de feromona con la probabilidad $(1-f)$. Como es mostrado en [11,18], el factor del ruido no debe exceder 5%, porque un factor de ruido mayor que 5% hace el sistema inestable.

Nosotros realizamos simulaciones y comparamos nuestro algoritmo con el enfoque propuesto en [10] usando la misma red SDH (*Synchronous Digital Hierarchy*). La red tiene 25 nodos parcialmente conectados y todos los enlaces tienen una capacidad de 40 llamadas. Además, se genera el tráfico de la

misma forma propuesta en la sección anterior, lo que determina también el número de hormigas requerido. Los fallos en los enlaces son generados aleatoriamente. Para este problema, los valores con los que se obtuvieron los mejores resultados para las variables α y β de la ecuación 1 son 0,2 y 0,6, respectivamente, y para ρ de la ecuación 2 es 0,5. Las Figuras 4 y 5 muestran estos resultados.

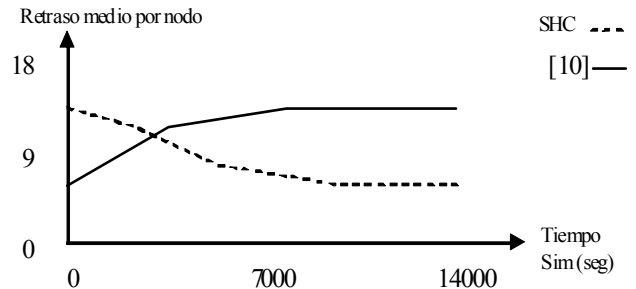


Figura 4. Retraso promedio por nodo

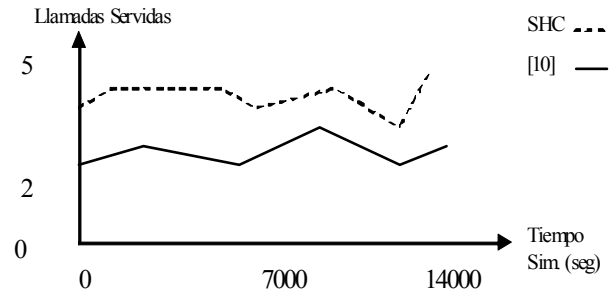


Figura 5. Tasa de transmisión

En la Figura 4 mostramos que nuestro enfoque proporciona un rendimiento mejor que [10] en la presencia de fallos. El retraso promedio por nodo es considerablemente mejor porque nosotros consideramos el problema de congestión de nodo. Igualmente, la Figura 5 muestra que la tasa de transmisión del sistema propuesto es buena, ya que maneja las variaciones de las llamadas entrantes y los fallos simultáneamente, mejorando a [10]. Nuestro algoritmo basado en agentes parece adaptarse particularmente bien a la topología de la red con constantes cambios. Esto realmente significa que el algoritmo de enrutamiento propuesto es un candidato bueno para redes con cambios en la topología constantes como las redes inalámbricas móviles del tipo ad hoc donde la movilidad de los nodos causa fallos de los enlaces constantemente. Nosotros también comparamos a nuestro modelo con el esquema descrito en [14] y la propuesta basada en Hormigas de [10] usando la misma red. En la Figura 6 se muestra que nuestro esquema proporciona un rendimiento mejor en tasa de transmisión en la presencia de fallos múltiples de enlaces/nodos.

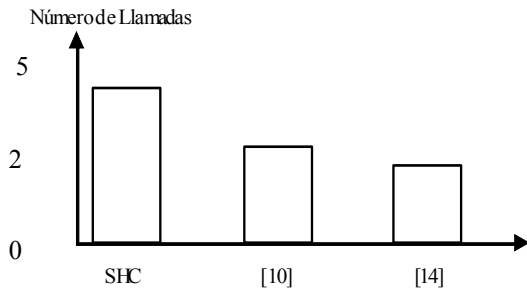


Figura 6. Tasa de transmisión

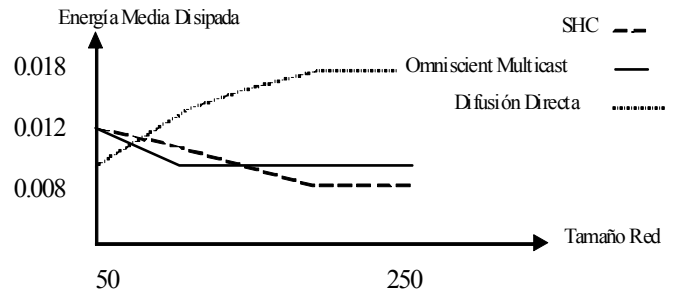


Figura 7. Promedio de energía disipada

C. Redes con restricciones de energía

Finalmente, también evaluamos nuestro algoritmo en las redes con restricciones específicas como las redes sensoriales inalámbricas con cambios topológicos, es decir con fallos en los nodos o enlaces. El problema de enrutamiento en las redes sensoriales inalámbricas es un gran reto por las capacidades de almacenamiento, procesamiento y de comunicaciones limitado de los nodos, así como el número grande de nodos. Como resultado, los aspectos principales de los protocolos de enrutamiento para las redes sensoriales inalámbricas son la eficiencia en el consumo de energía y la escalabilidad. Se han propuesto muchos protocolos de enrutamiento para las redes sensoriales inalámbricas [20, 21, 22, 25, 23, 25, 24, 26, 35].

Nosotros consideramos la métrica siguiente para analizar y comparar el rendimiento de los esquemas en nuestra evaluación. Primero, consideramos la media de energía disipada, la cual mide la proporción de energía disipada total por cada nodo en la red según el número de eventos distintos que suceden en ella (en datos en Joules/nodo/recibido). Esta métrica computa el trabajo promedio hecho por un nodo e indica la vida global del nodo sensorial. Nosotros también medimos el retraso promedio, o el tiempo promedio que toma transmitir un evento y recibirlo en cada sitio (en segundos). Las simulaciones se llevan a cabo con un número total de sensores que van de 50 a 250. El tráfico es fijo y consiste en cinco fuentes y cinco receptores. Se seleccionan las fuentes y receptores al azar, y se esparcen uniformemente en el campo sensorial. Cada fuente genera dos eventos por segundo (esto va determinando el número de hormigas (m) en un momento dado del algoritmo). Nosotros comparamos nuestro enfoque con [9], donde se proponen las técnicas denominadas como Difusión Dirigida y *Omnoscient Multicast*. Para este problema, los valores con los que se obtuvieron los mejores resultados para las variables α y β de la ecuación 1 son 0,6 y 0,3, respectivamente, y para ρ de la ecuación 2 es 0,6. La Figura 7 muestra la energía media disipada por un evento como una función del tamaño de la red, usando la ecuación 6. Nuestro modelo disipa un poco menos que la técnica *Omnoscient Multicast*, pero es mucho mejor que Difusión Dirigida. Para algunos campos sensoriales, nuestro modelo disipa sólo 60% de energía. Con respecto al retraso promedio observado (ver Figura 8), nuestro modelo tiene un retraso comparable a Difusión Dirigida. Así, nuestro modelo parece estar encontrando los caminos de retraso más bajos.

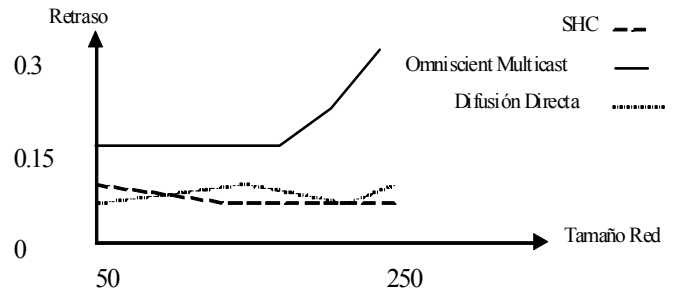


Figura 8. Retraso medio

Nosotros también realizamos las simulaciones con fallos en los nodos. Para cada campo sensorial, repetidamente, apagamos un fragmento fijo de los nodos durante 30 segundos. Estos nodos son uniformemente escogidos del campo sensorial. Mirando las Figuras 9 y 10, nuestro esquema mantiene la entrega de evento razonablemente mientras incurre en menos de un 20% de retraso promedio adicional, sobre todo en las redes más grandes. En cuanto a la energía, puede verse que la energía disipada no se ve afectada con la presencia de fallos porque nuestro modelo considera la formulación del problema de falla directamente (ver ecuación 5).

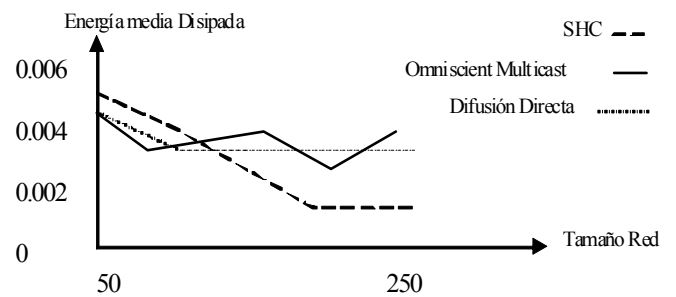


Figura 9. Promedio de energía disipada con fallos en la red

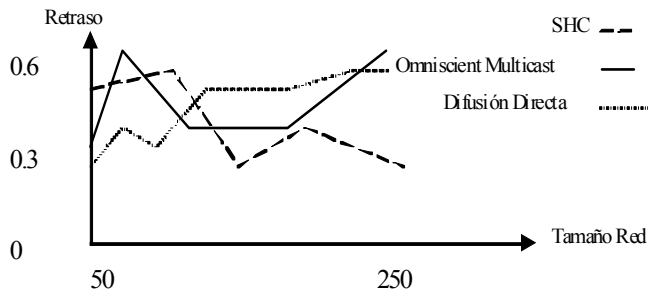


Figura 10. Retraso medio con fallos en la red

Más simulaciones fueron hechas para demostrar la sobrevivencia de la red. Como el trabajo hecho en [12], nosotros simulamos una red de 76 nodos, de los cuales 65 son sensores estáticos, 4 son sensores móviles, y 7 son controladores estáticos. 47 son sensores de nivel de líquido y 18 son sensores de temperatura. Los controladores envían demandas de datos a los sensores en su región de interés. Estas demandas programan a los sensores de nivel de líquido para enviar cada 10 segundos datos y a los de temperatura cada 30 segundos. Nosotros comparamos nuestro algoritmo con [12] y [9] y utilizamos la Ecuación 7 para la selección del camino con $\alpha=1$ y $\beta=5$. La red es simulada hasta que un nodo consume toda su energía. En el algoritmo propuesto en [9], esto ocurrió después de 150 minutos, mientras tomó 216 minutos en el caso de [7], y 250 minutos para nuestro protocolo. Esto es un aumento en la vida de la red de 64% con respecto al 44% de [7].

V. CONCLUSIONES

En este trabajo se propone un algoritmo de enrutamiento distribuido basado en el Sistema de Hormigas. El algoritmo propuesto muestra su generalidad y versatilidad en la posibilidad de utilizarse en diferentes redes de comunicaciones con alto rendimiento. Nuestro enfoque puede aplicarse a cualquier problema de enrutamiento definiendo una representación del grafo apropiada al espacio de solución del problema considerado, el procedimiento dinámico para actualizar esa representación, y una función objetivo que guíe nuestra heurística para construir soluciones factibles. En nuestro enfoque, el ambiente dinámico del problema de optimización combinatoria se define a través de la matriz del problema de Optimización Combinatoria, el cual genera el espacio que las hormigas atravesarán (grafo AS). Las hormigas atraviesan ese espacio según un conjunto de probabilidades que se van actualizando usando una regla de actualización de la feromona y la función objetivo del problema de optimización combinatoria considerado. Los mensajes entre los nodos son reemplazados por hormigas que simultáneamente actualizan los parámetros de la red y la feromona en la ruta entre la fuente y el destino. Los resultados muestran que nuestro enfoque obtiene un rendimiento bueno en la presencia de carga pesada y fallos múltiples (enlaces, nodos), contribuye a la minimización de la congestión (equilibrio de carga), y guarda estable la tasa de transmisión

de la red. Además, nuestro modelo puede usarse como un algoritmo distribuido con el potencial de economías significantes en el uso de energía, muy útil en las redes sensoriales inalámbricas.

Nuestro modelo requiere la definición de varias variables: α y β para la ecuación 1, ρ para la ecuación 2, y m (número de hormigas). Los resultados de nuestro modelo son sensibles a dichos valores, pero una vez obtenidos no deben ser modificados para el caso de estudio en particular (por ejemplo, para redes con restricciones de energía, los valores ideales son: $\alpha=0,6$, $\beta=0,3$, y $\rho=0,6$). Dichos valores dependen del comportamiento que uno desea que tenga el algoritmo, para el caso que se requiera explorar el espacio de soluciones α debe tender a 0, si se desea explotar las soluciones que se van obteniendo β debe tender a 0, si no se quiere tener memoria de los resultados que se van obteniendo ρ debe tender a 1. Por otro lado, el valor de m repercute en el tiempo de ejecución, es un valor dinámico que es función del tráfico existente en la red en un momento dado. Finalmente, según los resultados obtenidos, el comportamiento transitivo del algoritmo desde que se inicia hasta que empieza a obtener buenas rutas (fase inicial), en general, es relativamente pequeño. Esta fase de aprendizaje inicial es necesaria, tal que le permita al algoritmo caracterizar al ambiente (tráfico, distancias entre nodos, etc.). Esa caracterización se va ajustando a la dinámica del ambiente (fallos, cambio de tráfico, etc.) a través de los mecanismos de aprendizaje reforzado del algoritmo.

El uso de estos algoritmos en redes de telecomunicaciones reales es a nuestro entender, aun un paso por dar. Algunas de las razones de esto son: 1) Desconocimiento de cómo implementar eficientemente en entornos muy dinámicos a estos tipos de algoritmos distribuidos, y 2) Falta de pruebas prácticas para verificar que no solamente trabajan bien y dan los resultados esperados, si no que también tienen otras propiedades importantes para los proveedores de servicios como escalabilidad (trabajan bien en redes de comunicación de gran tamaño), fiabilidad, etc. Como se ha indicado en la sección III, a nivel de investigación hay un número importante de propuestas.

REFERENCIAS

- [1] J. Aguilar, L. Velásquez and M. Pool, "The Combinatorial Ant System." *Applied Artificial Intelligence*, Taylor and Francis, Vol. 18, N. 5, pp. 427-446, 2004.
- [2] J. Aguilar, "The Combinatorial Ant System for Dynamic Combinatorial Optimization Problems." *Revista de Matematica: Teoria y Aplicaciones*, Universidad de Costa Rica, San José, Costa Rica, 2005.
- [3] E. Bonabeau, M. Dorigo, G. Theraulaz, *Swarm Intelligence: from Natural to Artificial Swarm Systems*. Oxford University Press USA, 1999.
- [4] D. Corne, M. Dorigo, F. Glover, *New Ideas in Optimization*. McGraw Hill Holland, 1999.
- [5] M. Dorigo, "Optimization, Learning and Natural Algorithms". Ph.D Thesis, Politecnico de Milano, Italy, 1992.
- [6] M. Dorigo, V. Maniezzo, A. Coloni, "The Ant System: Optimization by a Colony of Cooperating Agents." *IEEE Transactions of Systems, Man, Cybernetics*, Vol. 26, pp. 29-41, 1996.

- [7] M. Dorigo, L. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem." *IEEE Transactions on Evolutionary Computation*, Vol. 1, pp. 53-66, 1997.
- [8] F. Hidrobo, J. Aguilar, "Toward a Parallel Genetic Algorithm Approach Based on Collective Intelligence for Combinatorial Optimization Problems." In *Proceedings of IEEE International Conference on Evolutionary Computation*, 1998, pp. 715-720.
- [9] C. Intanagonwiwat, R. Govindan, D. Estrin Direct, "Diffusion: A Scalable and Robust Communication Paradigm for Sensor," Networks. Technical Report, Defence Advanced Research Project DABT63-99-1-0011, 2002. Also, in *Proceedings of ACM MobiCom '00*, Boston, MA, 2000, pp. 56-67.
- [10] C. Mavromoustakis, H. Karatza, "Agent-Based Throughput Response in Presence of Node and/or Link Failure (on demand) for Circuit Switched Telecommunication Networks," *Computer Communications*, Vol. 27, pp 230-238, 2004.
- [11] R. Schoonderwoerd, O. Holland, J. Bruten, L. Rothkrantz, "Ant-based Load Balancing in Telecommunications Networks", *Adaptive Behavior*, Vol. 5, pp. 169-207, 1997.
- [12] R. Shah, J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks", Technical Report, DARPA Project F29601-99-1-0168, 2002.
- [13] C. Toh, "Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks." *IEEE Communications Magazine*, pp. 138-147, 2001.
- [14] D. Bertsekas, R. Gallager, *Data Networks*. Prentice Hall, New Jersey, 1987.
- [15] J. Aguilar, W. Ramirez, "Some Results of CAS for Ad Hoc Networks", Technical Report, CEMISID, Facultad de Ingenieria, Universidad de los Andes, 2004.
- [16] J. Bosh, D. Maltz, D. Johnson, Y. Hu, J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols." In *Proceeding of 4th Annual AAM/IEEE Intl. Conf. On Mobile Computing and Networking*, 1998.
- [17] W. Rabiner, A. Chandrakasan, H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks." In *Proceedings of the Hawaii Intl. Conference. on System Sciences*, 2000.
- [18] S. Appleby, S. Stewart, "Mobile Software Agents for Control in Telecommunication Networks." *BT Technology Journals*, Vol. 12, pp. 104-113, 1994.
- [19] R. Schoonderwoerd, O. Holland, J. Bruten, "Ant-like Agents for Load Balancing in Telecommunications Networks." In *Proceedings of Agents*, 1997, pp. 209-216.
- [20] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Ceyirci, "Wireless Sensor Networks: A Survey." *Computer Networks*, 38(4):393-422, 2002.
- [21] S. Hedetniemi, S. Hedetniemi, A. Liestman, "A Survey of Gossiping and Broadcasting in Communication Networks. *Networks*," Vol. 18, 1988.
- [22] W. Heinzelman, J. Kulik, H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks." In *Proceedings of ACM MobiCom '99*, Seattle, WA, 1999, pp. 174-85.
- [23] K. Sohrabi, J. Gao, V. Ailawadhi, G. J. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network," *IEEE Personal Communications*, Oct. 2000, pp. 16-27.
- [24] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks." *Proceedings of the 33rd International Conference on System Sciences (HICSS '00)*, January 2000, pp. 1-10.
- [25] S. Basagni, I. Chlamtac, V. Syrotiuk, B. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)." *Proceedings of MOBICOM*, pages 76--84, 1998.
- [26] G. Lukachan, M. Labrador, "SELAR: Scalable Energy-Efficient Location Aided Routing Protocol for Wireless Sensor Networks." *Proceedings of IEEE LCN Workshop on Wireless Local Networks (WLN) 2004*, pp. 694-695, Tampa, November 2004.
- [27] G. Caro, M. Dorigo, "AntNet: Distributed Stigmergetic Control for Communication Networks." *Journal of Artificial Intelligence Research*, 9:317-365, 1998.
- [28] E. Gabber, M. Smith, "Trail Blazer: A Routing Algorithm Inspired by Ants." *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP'04)*, 2004.
- [29] C. Mavromoustakis, H. Karatza, "Split Agent-based Routing in Interconnected Networks." *International Journal of Communication Systems*, Volume 17, Issue 4, May 2004, pp. 303-320.
- [30] H. Sandalidis, C. Mavromoustakis, P. Stavroulakis, "Ant Based Probabilistic Routing with Pheromone and Antipheromone Mechanisms." *International Journal of Communication Systems*, Volume 17, Issue 1, February 2004, pp. 55-62.
- [31] I. Stojmenovic, S. Datta, "Power and Cost Aware Localized Routing with Guaranteed Delivery in Unit Graph Based Ad Hoc Networks." *Wireless Communications and Mobile Computing*. Volume 4, Issue 2, March 2004, pp. 175-188.
- [32] C. Yu, B. Lee, H. Youn, "Energy Efficient Routing Protocols for Mobile Ad Hoc Networks." *Wireless Communications and Mobile Computing*, Volume 3, Issue 8, December 2003, Pages: 959-973.
- [33] C-Y. Chiu, G-H. Chen, EH-K. Wu, "A Stability Aware Cluster Routing Protocol for Mobile Ad Hoc Networks." *Wireless Communications and Mobile Computing*, Volume 3, Issue 4, June 2003, pp. 503-515.
- [34] A. Boukerche, L.Zhang, "A Performance Evaluation of a Pre-emptive On-demand Distance Vector Routing Protocol for Mobile Ad Hoc Networks." *Wireless Communications and Mobile Computing*, Volume 4, Issue 1, February 2004, pp. 99-108.
- [35] J. Aslam, O. Li, D. Rus, "Three Power-aware Routing Algorithms for Sensor Networks." *Wireless Communications and Mobile Computing*, Volume 3, Issue 2, March 2003, pp. 187-208.
- [36] C. Low, C. Yuen, "Energy Efficient Broadcast Routing in Mobile Ad Hoc Networks." *International Journal of Communication Systems*, Volume 17, Issue 4, May 2004, pp. 375-388.



Jose Aguilar (M'1999) obtuvo una Maestría en Informática en 1991 en la Universidad Paul Sabatier-Toulouse-France, y el Doctorado en Ciencias Computacionales en 1995 en la Universidad Rene Descartes-Paris-France. Además, realizó un Postdoctorado en el Departamento de Ciencias de la Computación de la Universidad de Houston entre 1999 y 2000.

El Dr. Aguilar es Profesor del Departamento de Computación de la Universidad de los Andes e investigador del Centro de Microcomputación y Sistemas Distribuidos (CEMISID) de la Universidad de los Andes y ha sido profesor/investigador visitante en varias universidades y laboratorios de prestigio en Francia, España y Estados Unidos. Sus áreas de interés son los sistemas paralelos y distribuidos, computación inteligente, optimización combinatoria, reconocimiento de patrones, y sistemas de control. Ha publicado varios libros en las áreas de Sistemas Computacionales y Gestión en Ciencia y Tecnología; y forma parte de varios comités editoriales de revistas. Él ha formado parte de varios jurados de premios científicos; ha presidido varios simposios, talleres, etc.; ha sido editor de varias actas de conferencias y de libros; y es revisor de revistas internacionales permanentemente. El Dr. Aguilar ha recibido varios premios nacionales como internacionales.



Miguel A. Labrador (M'1997, SM'2005) Obtuvo su Maestría y Doctorado en Telecomunicaciones de la University of Pittsburgh, en 1994 y 2000 respectivamente.

Antes de unirse a la facultad del Department of Computer Science and Engineering de la University of South Florida como profesor asistente en 2001, trabajó en Telcordia Technologies, Inc. en el grupo de Redes Banda Ancha de la organización de Servicios Profesionales. Sus intereses de investigación están en el diseño y evaluación del rendimiento de redes de computación y protocolos de comunicación para redes inalámbricas y ópticas.

El Dr. Labrador ha servido como miembro de los comités técnicos de muchas conferencias de la IEEE y es miembro de cuerpo editorial de "Computer Communications", Elsevier Science y "Revista Colombiana de Computación" Colombia. El fue el secretario de el IEEE Technical Committee on Computer Communications (TCCC) y el Chair del Symposium "Transport Layer Protocols over Wireless Networks" de la conferencia IEEE VTC 2003.