

# Estudio Experimental del Comportamiento de Algoritmos de Hormigas Aplicados a Enrutamiento en Redes de Computadoras

**Rolando F. Vallejo**  
ITESM - Campus Ciudad de México  
rolando.vallejo@itesm.mx

**Roberto Gómez C., Edgar E. Vallejo**  
ITESM - Campus Estado de México  
rogomez@itesm.mx, vallejo@itesm.mx

## Resumen

*Los algoritmos de enrutamiento se utilizan para encontrar caminos útiles en el establecimiento de comunicaciones eficientes en redes de computadoras. Conforme el tamaño de la red aumenta, encontrar esas rutas se torna más complejo en una manera no lineal. Con el afán de resolver problemas como éste, algunos investigadores han presentado técnicas innovadoras, tales como los algoritmos de hormigas. Este documento presenta un estudio empírico consistente en experimentación relacionada con el uso de esta clase de algoritmos aplicados al problema del enrutamiento, como un escenario para comprender su comportamiento. Los resultados mostrados en este artículo indican que algunos factores involucrados en el proceso pueden ser sintonizados para mejorar el rendimiento del algoritmo.*

## 1 Introducción

Algunos investigadores han estudiado el comportamiento de los insectos sociales. Las colonias de insectos muestran un alto grado de eficiencia en sus actividades y en sus capacidades de supervivencia. Se cree que la causa de esta eficiencia recae en el hecho de que, mientras que cada individuo de la colonia considera su propia labor como si éste se encontrara solo, el impacto de su actividad sobre la colonia parece ser parte de una coordinación centralizada.

La *estigmergia* † [GRA59] es un tipo de comunicación indirecta que involucra a los individuos de las colonias de insectos sociales. En el proceso de comunicación algunos de los sentidos no son necesarios para que esta tenga lugar. Se trata de una herramienta muy útil para coordinar y dividir los esfuerzos de grupos de individuos, y a esto se le atribuye el gran éxito que han mostrado las colonias para alcanzar sus objetivos primarios de supervivencia.

Durante la recolección del alimento, las hormigas depositan en el piso un rastro de feromona mientras se desplazan. Esta feromona actúa como un agente guía para los demás individuos, desde el nido hacia la fuente de alimento, sin necesidad del sentido de la vista. El rastro de feromona permanece en el piso por un periodo de tiempo limitado, y este hecho tiene repercusión directa sobre la concentración de la hormona. Cuando las hormigas llegan a la fuente de alimento y regresan sobre sus pasos hacia el nido, en caminos cortos se deposita feromona más frecuentemente que en los caminos largos, así que existe una mayor concentración de feromona cuando la ruta elegida es más corta.

† El término en inglés *stigmergy* parece no tener una traducción al español. Usaremos la palabra *estigmergia* para referirnos a *stigmergy* aunque ésta no existe en el Diccionario de la Real Academia Española.

Adicionalmente, tenemos que las hormigas prefieren moverse sobre caminos que tienen una mayor concentración de feromona. En consecuencia, las hormigas generalmente encuentran la ruta más corta hacia la fuente de alimento, aunque con cierta probabilidad, éstas proceden de manera inusual.

Mientras esto ocurre en el mundo natural, nos gustaría referirnos al problema de enrutamiento. El término *enrutamiento* se usa para referirse a la función principal de la Capa de Red del Modelo OSI, que consiste no sólo en encontrar un camino entre computadoras de una red, sino que también es muy importante encontrar el mejor de ellos, cuando existe más de una ruta posible para seguir hacia la computadora destino.

Conforme el tamaño de la red crece, el proceso de enrutamiento se vuelve más complejo, y algunos de los algoritmos implementados en los equipos de enrutamiento pueden no mostrar el mejor desempeño, de manera que el comportamiento distribuido de estos sistemas, es uno de los puntos que deben ser analizados sensiblemente.

Trabajos previos [DCG99] [DBT99] han mostrado que es posible modelar algunos problemas complejos, como Optimización Discreta, Asignación Cuadrática y Agente Viajero entre otros, inspirándose en la forma en que los insectos sociales resuelven sus necesidades.

El propósito de éste artículo es el de contribuir al entendimiento de los algoritmos de hormigas, de como funcionan, , y presentar resultados relevantes de un juego de experimentos empíricos, en la búsqueda de la combinación óptima de parámetros involucrados en el proceso. El algoritmo utilizado fue ANTNet, desarrollado por Dorigo et al [DCG99] [DBT99], cuya intención es la de aproximar el problema de optimización de determinación de rutas. Más que una simulación del proceso de enrutamiento y su solución, intentamos mostrar cómo el orden se genera en sistemas de hormigas.

Hemos encontrado un rango de valores conveniente para las variables del sistema, y creemos que éstos pueden ser utilizados en la mayoría de las topologías de red.

## 2 Trabajos Previos

Dorigo et al [DCG99] [DBT99] introdujo un algoritmo para enrutamiento adaptativo, basado en el comportamiento de las hormigas en su actividad de consecución de alimento. Estos algoritmos exploran redes de datos con el propósito de construir tablas de enrutamiento, y mantenerlas adaptadas a las condiciones de tráfico en la red. Este algoritmo recibió el nombre de *ANTNet*.

Fundamentalmente, el algoritmo consiste en la creación de hormigas artificiales en cada nodo de la red, representada por un grafo con nodos y enlaces bidireccionales, a intervalos de tiempo regulares.

- Cada nodo se caracteriza por el número de vecinos que tiene y por su tabla de enrutamiento, que contiene las probabilidades de elegir el vecino  $n$  como siguiente salto, siendo  $d$  el nodo destino.
- Las hormigas eligen aleatoriamente un nodo destino y seleccionan el siguiente salto entre los nodos vecinos no visitados previamente. La probabilidad de seleccionar un nodo particular, es proporcional al valor que tiene la tabla de enrutamiento para este nodo, así como una cola local, la cual es creada en cada nodo por el tráfico local.
- El identificador de cada nodo visitado se empuja en una pila que forma parte de la estructura de datos de la hormiga. El tiempo que le toma a cada hormiga para llegar al nodo destino también se almacena.

- Cuando una hormiga alcanza el nodo destino, ésta genera una *hormiga de regreso* y le transfiere el contenido de la pila a la hormiga nueva, y entonces ésta muere.
- La hormiga de regreso toma la misma ruta que su antecesora, pero en dirección contraria. Conforme la hormiga avanza hacia el nodo origen, actualiza las tablas de enrutamiento conforme se explica a continuación:
  - Cuando el tiempo utilizado por la hormiga para llegar al nodo destino fue menor que el tiempo memorizado por el modelo, entonces la hormiga actualiza la tabla de enrutamiento, en otras palabras, deposita mayor cantidad de feromona.
  - Cuando el tiempo utilizado por la hormiga para llegar al nodo destino fue mayor que el tiempo memorizado por el modelo, entonces la hormiga no efectúa cambio alguno en la tabla de enrutamiento..

El algoritmo de Dorigo's et al se comparó con los algoritmos de enrutamiento tradicionales:

- Una versión simplificada de *Open Shortest Path First (OSPF)*, el algoritmo oficial para Internet.
- Una versión sofisticada del algoritmo asíncrono y distribuido conocido como *Bellman Ford (BF)*, con una métrica dinámica de costos.
- El algoritmo *Shortest Path First (SPF)*, con una métrica dinámica de costos.
- El algoritmo *Q-Routing* propuesto por Boyan and Littman.
- El algoritmo *Predictive Q-Routing*, una extensión del algoritmo *Q-Routing*.

Los parámetros observables en las pruebas fueron:

- El retardo en la entrega de los paquetes.
- El tráfico total transportado, conocido como *throughput*.

Con respecto al tráfico total transportado, las variaciones entre los diferentes algoritmos fueron mínimas, mientras que para el retardo, *ANTNET* mostró un rendimiento 4 veces mejor que *Q-routing*, 3 veces mejor que *Predictive Q-Routing*, 2.5 veces mejor que *OSPF* y 1.5 veces mejor que *BF* y *SPF*.

Creemos que el rendimiento del algoritmo puede ser optimizado mediante la apropiada manipulación de los parámetros involucrados. La manipulación dinámica de estos parámetros es parte de nuestro modelo propuesto.

Otros trabajos relevantes han sido desarrollados por científicos como Schoonderwoerd et al [SHB96], White et al [WPO98], Heusse et al [HGS00], y podrían ser interesantes para el lector.

### 3 Modelo

El algoritmo propuesto consiste en la creación de agentes artificiales (hormigas y feromona) los cuáles simulan los factores que actúan en el proceso de estigmergia. Estos agentes están conformados por una colección de parámetros que pueden ser variados cuantitativamente para observar el desempeño global del sistema, e intentar encontrar la combinación óptima de valores que ofrezcan la aproximación al problema en el menor tiempo posible.

Encontrar los mejores valores para estos parámetros no es una tarea trivial, ya que el comportamiento global del sistema es una propiedad emergente de la interacción local de los agentes individuales, en consecuencia, este comportamiento no puede ser explicado a partir de las capacidades individuales de los agentes.

El modelo considera un ambiente simulado de redes de área amplia (WANs), conformadas por nodos en un grafo, y cada nodo representa una red de área local (LAN). Los enlaces entre los nodos son representados por lados en el grafo.

El algoritmo consiste en la creación de una hormiga para cada nodo de la red, con un destino elegido al azar. Esta primera generación de hormigas elige el siguiente nodo a visitar en la búsqueda del nodo destino, utilizando una distribución uniforme de probabilidades.

Durante la selección del siguiente nodo, es necesario verificar que el nodo elegido no haya sido previamente visitado por la misma hormiga, esto con el fin de evitar que la hormiga caiga en un ciclo infinito. Para lograr esto, la hormiga almacena el identificador de cada nodo que visita.

Si la hormiga llega a un callejón sin salida, es decir, todos los nodos vecinos han sido visitados previamente, entonces la hormiga muere y una nueva hormiga es creada en ese nodo, nuevamente, con un destino nuevo seleccionado aleatoriamente.

Cada nodo tiene una tabla de enrutamiento asociada a él, que representa la concentración de feromona que existe en ese nodo.

Esta tabla se conforma por una arreglo bidimensional cuyos valores se definen por una función probabilística  $P_{nd}$ . Esta función es la probabilidad de elegir el enlace que lleva a la hormiga hacia el nodo vecino  $n$  cuando el nodo destino es  $d$ . En otras palabras,  $P_{nd}$  es el grado de *bondad* de esta ruta hacia el nodo destino  $d$ . Inicialmente, esta tabla tiene valores que corresponden a una distribución uniforme de probabilidades.

La operación del sistema se ejecuta en pasos discretos. En cada paso, todas las hormigas avanzan un nodo. El nacimiento de una nueva generación de hormigas se hace de manera iterativa, después de cierto número de pasos, y la frecuencia de nacimientos es un parámetro controlable.

Cuando una hormiga alcanza su destino, regresa al nodo origen depositando una cantidad de feromona en cada nodo conforme regresa. Esta acción se ejecuta en función de la cantidad de energía que la hormiga tiene aún en su cuerpo; si la hormiga tuvo que moverse a través de más nodos, posee menos energía y menos feromona para depositar.

Cuando una hormiga llega a un nodo en particular, encuentra que la tabla de enrutamiento ha sido modificada por otras hormigas. De esta manera aparece la comunicación indirecta entre individuos, la cuál es referida como estigmergía.

En la última generación, creamos tantas hormigas como nodos existen en el sistema. Se intenta llegar desde ese nodo hasta todos los demás, con el fin de evaluar el trabajo de todas las generaciones anteriores de hormigas seleccionando de manera determinista la liga con mayor concentración de feromona, justamente como lo habría hecho un paquete de datos, cuyo comportamiento no sería necesariamente estocástico.

Para simular cierta sensibilidad de las hormigas a la feromona, buscamos una variable existente en el sistema. La variable seleccionada fue el número de hormigas vivas en cada paso, como un factor que modifica dinámicamente la cantidad de feromona en el sistema para evaluar el rendimiento del algoritmo.

Lo que se busca, es que un gran número de hormigas actuando sobre el sistema, eventualmente generen rutas con el número mínimo de saltos, sin importar el nodo origen y destino para cualquier paquete de datos.

## 4 Experimentos y Resultados

Es necesario mencionar que el parámetro seleccionado para evaluar el sistema es *conteo de saltos*, ó el número de nodos que una hormiga visita hasta que alcanza el destino. La razón por la que escogimos esta métrica es porque algunos de los algoritmos de enrutamiento, como algoritmos por vector de distancias, usan esta métrica. Bajo este esquema, una ruta con menos saltos es considerada mejor que otra en la que se requiera que el paquete de datos viaje a través de una cantidad mayor de nodos.

Una de las intenciones de este trabajo es la de tener la posibilidad de validar el modelo con instancias representativas del problema. Hemos utilizado para la experimentación dos topologías de red distintas, conocidas como *NSFNet* (Figura 1) y *NTTNet* (Figura 2), cada una de ellas con enlaces bi-direccionales.

La razón por la que seleccionamos éstas instancias de redes, fue porque la primera representa una red de complejidad típica, y la segunda es una red balanceada (en número de nodos y enlaces).

Por otro lado, estas topologías fueron utilizadas también por Marco Dorigo en sus experimentos, y consideramos importante encontrar coincidencias y desviaciones entre nuestro trabajo y el de Dr. Dorigo.

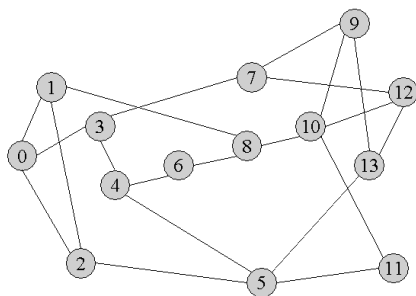


Figura 1. NSFNet

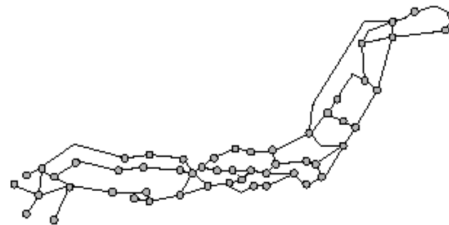


Figura 2. NTTNet

### 4.1 Configuración de los Parámetros del Sistema

Una parte importante de la etapa de experimentación es la selección de los parámetros del sistema. Alguno de ellos fueron manipulados de manera sistemática, tratando de encontrar correlación entre esos parámetros y el impacto en el rendimiento global del algoritmo, ejecutando un análisis de sensibilidad.

Los parámetros a manipular fueron: El número de pasos que el sistema actúa, la frecuencia de nacimiento de nuevas generaciones de hormigas, y la cantidad de feromona de cada hormiga.

Pruebas preliminares mostraron que los valores representativos para ejecutar las diferentes modalidades desarrolladas del programa, fueron aquellas que se muestran más adelante. Esas ejecuciones del programa fueron efectuadas con cada posible combinación de valores mostrados más adelante, aproximadamente 2,700 ejecuciones en total:

- Número de pasos del sistema: 2048, 4096, 8192, 16384, 32768, 65536
- Frecuencia de nacimiento de nuevas generaciones de hormigas: 64, 32, 16
- Cantidad de feromona: 0.001, 0.005, 0.01, 0.05, 0.1

Pensamos que estos valores conforman un conjunto representativo de pruebas que nos permiten distinguir y analizar la incidencia de los factores involucrados y las diferencias entre las versiones del programa.

#### 4.2 Análisis y Discusión de los Resultados

En esta sección presentamos algunos ejemplos representativos de ejecuciones del programa y sus resultados, mismos que nos llevaron a establecer las conclusiones más importante de nuestro trabajo.

La mayoría de las pruebas fueron efectuadas sobre *NSFNet*, ya que la consideramos una red de complejidad típica y está bien balanceada. Otra razón fue que el tamaño de la red nos permitió calcular manualmente los resultados óptimos, para poder comparar éstos con la salida del sistema.

#### 4.3 Parámetros Observables

Como consideramos una métrica de saltos, pensamos que los parámetros más importantes a medir son:

- La suma total de saltos que todas las hormigas usaron durante la última generación. Si el sistema funciona correctamente, este valor debe ser cercano a 278 saltos totales, que es la medida óptima del sistema. En adelante, a este parámetro lo Llamamos *Global*.
- Promedio de saltos para cada hormiga de la última generación; el valor óptimo calculado es 3.15 saltos por hormiga. En adelante, a este parámetro lo Llamamos *Promedio*.
- Por último, consideramos la cantidad total de hormigas que fallaron en la búsqueda del nodo destino; cuando las tablas de enrutamiento no han convergido en valores bien definidos, las hormigas vagan sin encontrar el nodo destino, entran en callejones sin salida y en consecuencia mueren. En adelante llamaremos a este parámetro *fallas*.

#### 4.4 Análisis de los Parámetros Controlables

Como se expone antes, los parámetros manipulados fueron:

- Cantidad de pasos que el sistema actúa. En adelante *pasos*.
- Frecuencia de nacimiento de generaciones de hormigas, expresada en cantidad de pasos, es decir, una frecuencia de 16 significa que una generación nueva de hormigas nace cada 16 pasos. Llamamos a esta variable *frecuencia*.
- Cantidad de feromona. La llamamos *feromona*.

Los resultados muestran una importante relación entre los tres diferentes parámetros controlables. Algunas combinaciones de valores producen un comportamiento del sistema que muestra resultados de parámetros observables muy cercanos al óptimo, mientras que otras combinaciones produjeron muchas fallas ó rutas muy largas.

Una observación muy importante es que, cuando parece que variar un parámetro no tiene ningún efecto sobre el rendimiento global, de pronto se aprecian resultados muy deseables al efectuar una pequeña modificación de alguna cantidad.

En trabajos previos con comportamiento emergentes [MIL94], se ha observado que “*el orden se genera en la frontera del caos*”. En otras palabras, cuando parece que el sistema actúa de manera errática, de pronto se estabiliza, lo cuál nos permite pensar que existe una línea muy fina que separa los buenos de los malos resultados, y en algunas ocasiones, los cambios pueden ser abruptos.

Abajo analizamos cada uno de los parámetros controlables y su incidencia sobre el rendimiento global.

#### 4.5 Pasos

Los resultados de los experimentos indican que la cantidad de pasos que el sistema actúa es un factor muy importante para el desempeño global del algoritmo. Para valores similares de *frecuencia* y *feromona*, mientras incrementamos el número de pasos, observamos que el parámetro *global* desciende substancialmente, como se muestra en la figura 3.

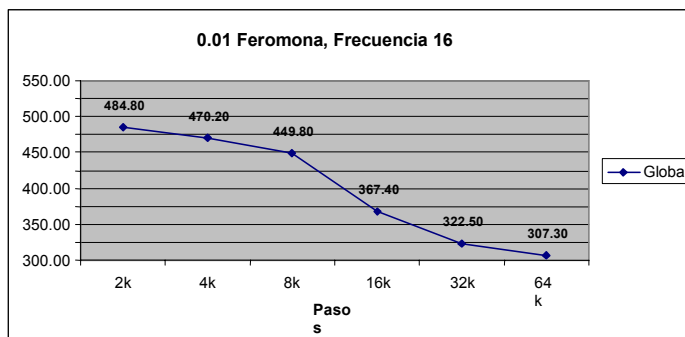


Figura 3: 0.01 feromona, frecuencia 16

Para valores similares de *frecuencia* y *feromona* observamos una valor de *global* muy cercano al valor óptimo (278), independientemente del valor manejado en *pasos*, como se muestra en la figura 4.

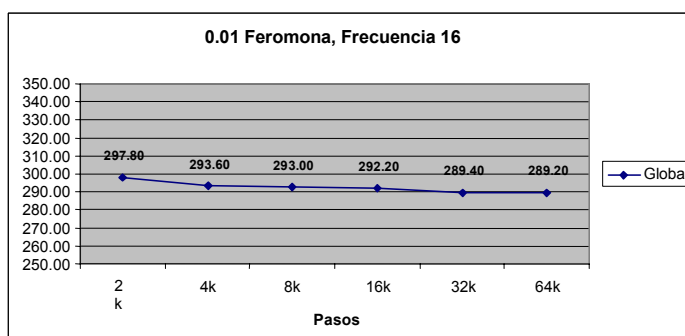


Figura: 4 0.01 feromona, frecuencia 16

#### 4.6 Frecuencia

Incrementar la frecuencia de nacimientos de hormigas, tiene un efecto similar sobre el sistema que incrementar el número de pasos, pero el tiempo necesario para tener una convergencia del sistema es menor. Al final, la relación entre *pasos* y *frecuencia* tienen efecto sobre la cantidad de hormigas vivas en el sistema. Observamos una cantidad similar de *fallas* y *promedio* para valores opuestos de

*frecuencia* y *pasos*, pero valores similares de *feromona*, presentaron *pheromone*, presentaron salidas similares.

Para valores similares de parámetros controlables que el ejemplo anterior, los valores de *promedio* no cambian considerablemente cuando se incrementa la cantidad de *feromona* y como podemos observar en la figura 5, este se acerca a valores óptimos en la mayoría de los casos, sin importar cuanta *feromona* se considere en el experimento. Por otro lado, la cantidad de *fallas* es prácticamente igual a cero para cualquier valor de *feromona*.

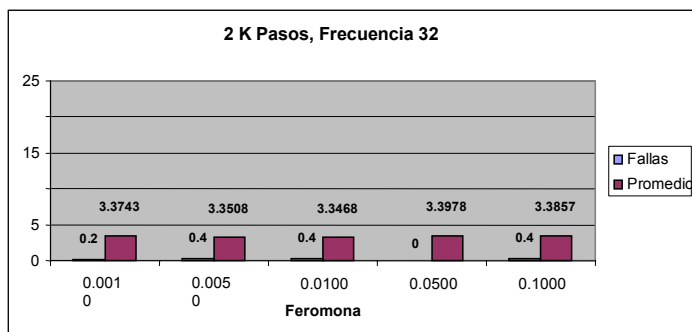


Figura 5: 2K pasos, frecuencia 32

#### 4.7 Feromona

La cantidad de *feromona* asignada a cada hormiga juega un importante papel en el desempeño del algoritmo. Si cada hormiga tiene suficiente energía para depositar un rastro más intenso de la hormona, entonces menos hormigas serán necesarias para construir tablas de enrutamiento convenientes.

Observamos en la experimentación que la cantidad de *feromona* tiene un impacto importante sobre *fallas* y *promedio* para cualquier variación de otros parámetros controlables. Esto se muestra en la figura 6.

Para ilustrar la importancia de la aseveración de Millonas [MIL94], “*El orden se genera en la frontera del caos*”, mostramos la figura 7, en donde podemos observar que usando cantidades fijas de *feromona* y *frecuencia*, pasar de 48 k *pasos* a 8 k *pasos* la variable *global* desciende notablemente. En la figura 8 observamos que con una cantidad fija de *pasos* y *frecuencia*, un pequeño incremento de *feromona* es suficiente para llevar el valor de *fallas* de 4.7 a 0, mientras que *promedio* baja de 5.21 a 3.58.

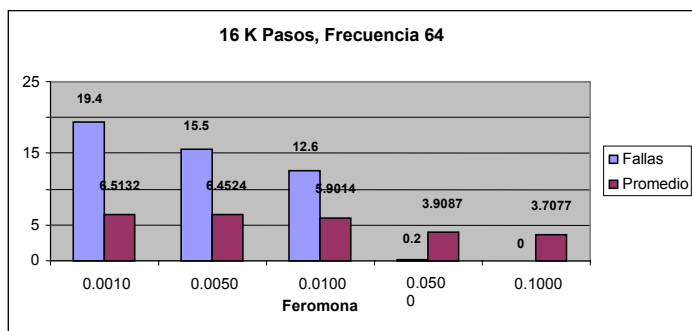


Figura 6: 16K pasos, frecuencia 64



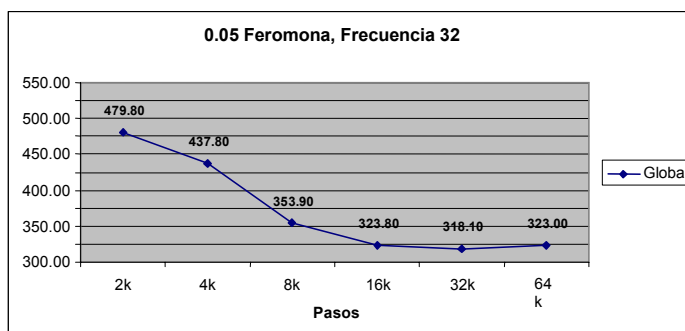


Figura 7. Pasando de 4K a 8K pasos, global desciende de manera importante

La combinación de los mejores valores para los parámetros controlables, depende fundamentalmente del tamaño de la red. Para una red con menor cantidad de nodos, usualmente los valores que arrojan mejores resultados son los siguientes:

- Frecuencia 32 ó 16
- 8K pasos o mayor
- 0.005 feromona o mayor

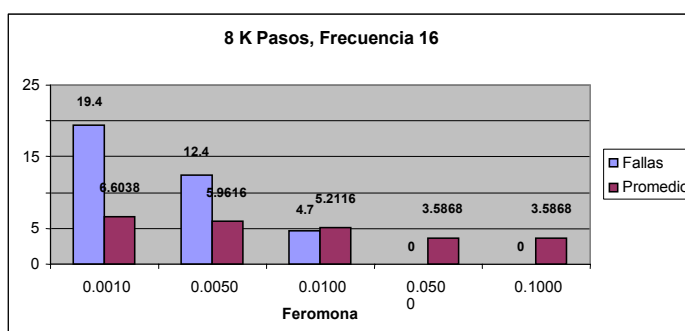


Figure 8: Pasando de 0.01 a 0.05 feromona, fallas y promedio descienden de manera importante

Para una red de tamaño medio, los valores recomendados son:

- Frecuencia 32 ó 16
- 32K pasos o mayor
- 0.01 Feromona o mayor

Para una red de tamaño mayor, los valores recomendados son:

- Frecuencia 16
- 64K pasos o mayor
- 0.05 feromona o mayor

Es importante mencionar que una vez que los resultados se acercan a valores óptimos, incrementar los valores de las diferentes variables controlables, no nos lleva a un mejor resultado aún. Creemos que el sistema alcanza cierta estabilidad en un punto en donde los parámetros tienen un valor

conveniente, y cualquier variación en la combinación no contribuye a un mejor desempeño del algoritmo.

#### 4.8 Resultados en NTTNet

Como se menciona antes, la mayoría de los experimentos fueron ejecutados sobre *NSFNet*. Hicimos algunos otros experimentos sobre *NTTNet* y éstos son los comentarios al respecto:

##### 4.8.1 *NTTNet*

En el caso de esta red, no calculamos manualmente los valores óptimos para *global* y *promedio*, debido a la alta complejidad que la instancia del problema presentaba. Sin embargo, creemos que es importante incluir nuestros resultados para su comentario.

En general, obtuvimos valores para *promedio* de 8.5 cuando no existen *fallas*. Asumimos que este valor es conveniente, cuando estamos hablando de una red con 55 nodos. La variable *global* encuentra una ligera variación negativa conforme *pasos* es incrementado, lo que nos lleva a hacer una conclusión importante: la cantidad de *pasos* que el sistema requiere es más grande que conforme el número de nodos crece.

El tamaño de la instancia del problema podría hacer inmanejable el uso de métodos convencionales, si intentáramos encontrar las mejores rutas en una manera no distribuida.

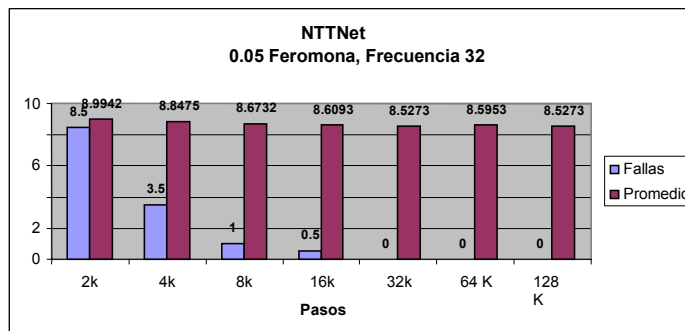


Figura 9: *NTTNet*, 0.05 feromona, frecuencia 32

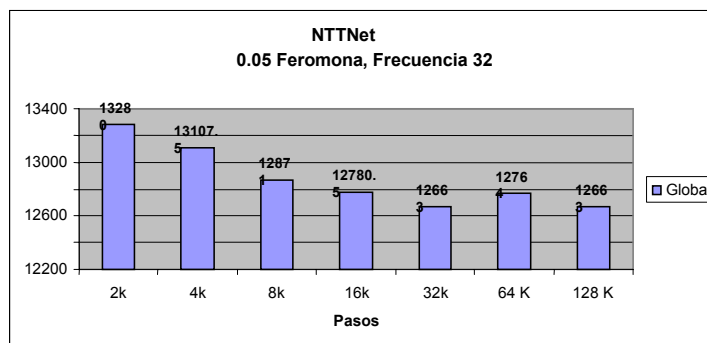


Figura 10: *NTTNet*, 0.05 feromona, frecuencia 32

## 5 Conclusiones y Trabajos Futuros

El uso de algoritmos inspirados en el comportamiento de insectos sociales, han reportado resultados promisorios en la búsqueda de soluciones a problemas complejos. El enrutamiento en redes con algoritmos distribuidos de hormigas, representan un nuevo y excitante tópico para investigación, porque creemos que el conocimiento en *swarm intelligence* [DCG99] [DBT99], hasta ahora, ha sido adquirido de manera fragmentada, y es tiempo de unir todas las piezas.

Concluimos que la simplicidad de la implementación del algoritmo presentado, juega un rol importante para alcanzar los objetivos. Nuestro trabajo en comparación con el trabajo de Dorigo, ofrece una ventaja: podríamos resaltar que la implementación es más simple, pero captura la esencia del problema y detecta las variables que participan en el comportamiento global del sistema, sin la necesidad de una complicada implementación del algoritmo.

Hemos encontrado un rango conveniente de valores para las variables del sistema, y creemos que estos valores pueden ser utilizados en la mayoría de las topologías de red que pudieran presentarse; sin embargo, sería necesario efectuar más pruebas para declarar esto con mayor confiabilidad.

Confirmamos con experimentación la aseveración hecha por Mark M. Millonas [MIL94]. Cuando parece que el sistema global se comporta de manera errática, de pronto se estabiliza, lo cuál nos deja pensar que existe una línea muy delgada que divide buenos de malos resultados, y en ciertas situaciones, los cambios pueden ser abruptos, y es por eso que la selección correcta de los parámetros óptimos es aún más importante.

Aún cuando estas técnicas de enrutamiento distribuidas podrían ser vistas como modernas y prometedoras, no han sido implementadas aún en contextos de redes reales, porque ciertos aspectos importantes deben ser antes tomados en cuenta. Los modelos de red usados hasta ahora, han sido simplificaciones de la realidad, y requieren ser probados en situaciones más reales:

- Considerando control de tráfico y congestión.
- Control de flujo.
- Calidad de servicio
- Pérdidas de paquetes.
- Requerimientos de ancho de banda.
- Tablas de enrutamiento en función de otras métricas y costos.

Los resultados obtenidos en éste trabajo, podrían ser utilizados como punto de partida para trabajos futuros.

En redes con un gran número de nodos, la ejecución del programa podría tomar algunas horas, dado el número de agentes independientes que se encuentran activos en cada etapa de la ejecución, y es recomendable implementar el algoritmo como un sistema distribuido.

## Referencias

- [DCG99] Dorigo, M., Di Caro, G., Gambardella, L., Ant Algorithms for Discrete Optimization. *Artificial Life, The MIT Press. Volume 5, Number 5, 1999.*
- [DBT99] Dorigo, M., Bonabeau, E. Theraulaz, G., Swarm Intelligence, From Natural to Artificial Systems. The Santa Fe Institute Studies in the Sciences of Complexity. 1999.

- [DAG90] Deneubourg, J., Aron, S., Gross, S., Pasteels, J.M., The Self-Organizing Exploratory pattern of the Argentine Ant. *Insect Behavior*. 1990.
- [GRA59] Grassé, P., La Reconstruction du nid et les Coordinations Inter-Individuelles chez *Bellicositermes Natalensis* et *Cubitermes* sp. La Théorie de la Stimergie: Essai d'interprétation du Comportement des Termites Constructeurs *Insect. Soc.* 6 1959.
- [HGS00] Heusse, M., Guerin, S., Snyers, D., Kuntz, P., Adaptive Agent-Driven Routing and Load Balancing in Communication networks, Accepted for publication in the *Journal of Complex Systems*. 2000.
- [MIL94] Millonas, M., Swarms, Phase Transitions, and Collective Intelligence. *Artificial Life III*, The Santa Fe Institute Studies in the Sciences of Complexity. 1994.
- [SHB96] Schoonderwoerd, R., Holland, O., Bruten, J., Rothkrantz, L., Ant-based load balancing in telecommunications networks. *Adaptive Behavior* 5 (1996)
- [WPO98] White, A., Pagurek, B., Oppacher, F., Connection Management using Adaptive Mobile Agents *Int. Cont. on Parallel Distributed Processing Techniques and Applications (PDPTA)* 802-809. CSREA Press (1998)