TACS, un Modelo de Confianza basado en Colonia de Hormigas

Félix Gómez Mármol felixgm@dif.um.es

MEMORIA TESIS DE MASTER

Máster en "Tecnologías de la Información y Telemática Avanzadas" – Curso 2006/07
Dpto. Ingeniería de la Información y las Comunicaciones
Dpto. Ingeniería y Tecnología de Computadores
Facultad de Informática. Universidad de Murcia.
Campus de Espinardo. 30100 Murcia. Spain.

D. Gregorio Martínez Pérez, Profesor Contratado Doctor del Área de Ciencia de la Computación e Inteligencia Artificial en el Departamento de Ingeniería de la Información y las Comunicaciones de la Universidad de Murcia, AUTORIZA:

La presentación de la Tesis de Máster titulada "TACS, un Modelo de Confianza basado en Colonia de Hormigas", realizada por D. Félix Gómez Mármol, bajo mi inmediata dirección y supervisión, en el Departamento de Ingeniería de la Información y las Comunicaciones.

En Murcia, a 25 de junio de 2007

Fdo. D. Gregorio Martínez Pérez

Resumen Extendido

Las redes P2P se encuentran en un momento de auge, teniendo detrás de ellas a muchos investigadores con el propósito de mejorarlas y perfeccionarlas. En este sentido, una de las principales líneas de investigación actuales en el campo de las redes P2P consiste en la búsqueda de modelos de confianza y reputación que permitan a un nodo cualquiera decidir si puede confiar o no en otro nodo de la red.

En este sentido, este trabajo pretende presentar un modelo de confianza original y novedoso que, haciendo uso de técnicas algorítmicas bioinspiradas sea capaz de reflejar adecuadamente la confianza que un nodo de la red merece de otro.

En concreto se ha usado una adaptación hecha por nosotros mismos del Sistema de Colonia de Hormigas (en inglés, *Ant Colony System*, o ACS). Mediante dicha adaptación se ha conseguido optimizar el comportamiento del algoritmo y con ello los resultados obtenidos.

Y se han obtenido resultados porque el modelo ha sido desarrollado desde sus primeras etapas de concepción, pasando por un diseño minucioso, hasta su eficiente implementación y puesta en marcha. Esto nos ha permitido llevar a cabo tanto análisis teóricos a priori, como análisis de resultados tangibles tras la ejecución del modelo ya implementado.

En primer lugar en este trabajo se verá una breve introducción del modelo de confianza basado en colonia de hormigas que hemos planteado. A continuación se expondrá una pequeña muestra representativa del abundante trabajo previo que sobre la materia en cuestión ya existe hoy en día.

El tercer apartado hablará del modelo en sí, explicándolo con detalle. Para ello primero se describirá brevemente el Sistema de Colonia de Hormigas tradicional en el cual nos basamos inicialmente para construir nuestro modelo. Una vez se hayan expuesto las nociones básicas sobre optimización basada en colonia de hormigas se estará en disposición de entender la descripción detallada y en profundidad del modelo. Después se mostrarán cuáles han sido las adaptaciones llevadas a cabo sobre el algoritmo tradicional de colonia de hormigas, argumentando cada una de ellas.

La última sección de este tercer apartado describirá con detalle las pruebas llevadas a cabo para examinar la calidad del modelo, con los resultados obtenidos, así como un análisis riguroso de los mismos. Con el objetivo de abarcar todas las situaciones que pudieran suceder en un entorno real decidimos plantear cuatro escenarios de pruebas distintos para observar el comportamiento de nuestro modelo frente a cada uno de ellos. Como se verá en dicha sección, el modelo arroja unos resultados más que satisfactorios en todos los casos.

Por último, unas conclusiones junto con algunas propuestas de vías y trabajo futuro cerrarán el documento, no sin antes detallar toda la bibliografía empleada en la elaboración del mismo.

Palabras clave: Confianza, Colonia de Hormigas, Red P2P, Cliente-Servidor.

1. Introducción

El reciente y veloz desarrollo de las redes P2P ha acaparado la atención de muchos investigadores por sus múltiples características ventajosas. Sin embargo, también adolecen de ciertas deficiencias que aún no han sido del todo resueltas. Una de estas deficiencias es la gestión eficiente y eficaz de la confianza en los nodos que componen la red P2P.

Un nodo de una red P2P podría no proporcionar los servicios que oferta cuando éstos le son requeridos, o incluso podría proporcionar servicios maliciosos con la intención de obtener un mayor beneficio individual.

Puesto que en este tipo de entornos no existe una entidad centralizada que supervise el comportamiento de los nodos, penalizando a aquellos que no se comporten adecuadamente, éstos últimos pueden campar a sus anchas sin ser castigados por ello.

Cómo distinguir nodos potencialmente benévolos de nodos potencialmente maliciosos es una de las principales líneas de investigación actuales en redes P2P. Si todo este tipo de problemas estuvieran resueltos, los sistemas P2P aumentarían sus aplicaciones enormemente. Hoy por hoy, el principal método para resolver estos problemas consiste en buscar mecanismos para calcular los valores de confianza y reputación en un sistema.

Muchas veces estos mecanismos están basados en los modelos de confianza y reputación entre humanos, apareciendo conceptos como “amigo”, “conocido”, “honradez”, “honestidad”, “traición”, etc. En nuestro caso entenderemos por confianza la esperanza que un nodo de la red tiene en que otro proporcione un determinado servicio con ciertos parámetros de calidad.

El modelo que nosotros presentamos es un modelo de confianza novedoso en cuanto a que está basado en un algoritmo de optimización bioinspirado. Concretamente, está basado en el Sistema de Colonia de Hormigas. Dicho algoritmo tiene una reputación ampliamente reconocida en el mundo de la computación evolutiva, ya que permite obtener unos de los mejores resultados en este campo.

Este modelo es de los primeros en la aplicación de dicho algoritmo bioinspirado en la forma en que nosotros lo hemos hecho. Tan sólo hemos encontrado otro modelo ligeramente parecido que también usa colonia de hormigas. Nuestro modelo, sin embargo y como veremos más adelante, es más completo y genérico que éste, además de imponer menos restricciones.

A la hora de diseñar el modelo pensamos que un modelo de confianza que supiera utilizar de forma eficaz y eficiente el algoritmo de colonia de hormigas sería capaz de obtener resultados de una gran calidad. Para ello debería saber explotar al máximo las capacidades y virtudes de este algoritmo bioinspirado.

Así por ejemplo, en un entorno tan dinámico y cambiante como es una red P2P donde los nodos pueden entrar y salir de la misma en cualquier momento y sin previo aviso, usar un sistema de colonia de hormigas pensamos que sería una opción inteligente. Las hormigas, en su búsqueda del mejor camino entre el hormiguero y la comida son capaces de buscar rutas alternativas si en cualquier momento se presenta un obstáculo en la ruta que actualmente tuvieran.

Además, las hormigas se van guiando por un rastro de feromona que ellas mismas van depositando por donde van pasando, de tal forma que aquel camino que más hormigas sigan tendrá un mayor rastro de feromona y así al final todas las hormigas acabarán convergiendo y siguiendo el mejor camino, el óptimo.

Nosotros decidimos identificar la feromona depositada entre dos nodos de un grafo (idénticamente, entre dos nodos de una red P2P) con la confianza que uno de ellos tiene en el otro, a la hora de proporcionar un determinado servicio en concreto. De esta forma, las hormigas nos permitirían encontrar aquel servidor de entre todos los nodos de la red P2P que mayor confianza mereciera a la hora de ofrecer un servicio en particular.

Finalmente lo llevamos a cabo y comprobamos que efectivamente se obtenían unos resultados altamente satisfactorios. Las hormigas eran capaces de detectar aquellos nodos que realmente merecían confianza e incluso distinguirlos de otros que no la merecían en absoluto. Es más, eran capaces de adaptarse a cambios en el comportamiento de los nodos, penalizando a aquellos que traicionaban su confianza y premiando a aquellos que ofrecían los mejores servicios.

2. Estado del arte

Los modelos de confianza en redes P2P han captado la atención de numerosos grupos de investigación. No en vano, han sido objeto de un estudio intensivo, dando como resultado bastantes modelos de confianza de una calidad considerable.

Así por ejemplo, en [1] los autores proponen *Sporas* como un mecanismo de reputación en sistemas de agentes donde la reputación se calcula recursivamente y en el que las valoraciones más recientes tienen más peso que las más antiguas.

El modelo propuesto en [2], llamado *Regret*, sugiere entender la reputación desde tres dimensiones distintas: la individual, dada de las interacciones directas con el agente, la social, de las experiencias previas de los miembros del grupo con el agente y sus conocidos, y la ontológica, dada de la combinación de múltiples facetas para construir una reputación sobre conceptos complejos.

Una mejora de este último modelo propuesta en [3] (*SocialRegret*) incorpora técnicas y mecanismos del análisis de redes sociales para modelar mejor la reputación social de un agente en un sistema multiagente.

Los autores de *Afras* [4] proponen un sistema de reputación en sistemas multiagente cuya principal característica reside en que modela tanto la reputación de un agente, como la evaluación de una transacción o interacción entre agentes, mediante conjuntos difusos.

Por su parte, en [5] se postula un modelo que calcula la confianza entre dos pares buscando todos los caminos que los conectan; a continuación para cada camino multiplica las recomendaciones asociadas a cada arista y por último agrega todos los caminos (normalizando dicha agregación).

En [6] se crea un vector de confianzas totales para cada nodo i, $\overrightarrow{TT_i} = \alpha \cdot \overrightarrow{DT_i} + (1-\alpha) \cdot \overrightarrow{Rep_i}$, siendo $\overrightarrow{Rep_i}$ la suma de las confianzas indirectas $\overrightarrow{Rep_i} = \overrightarrow{IDT_i}^1 + \overrightarrow{IDT_i}^2 + \cdots + \overrightarrow{IDT_i}^m$ y donde IDT_{ij}^m representa aquel camino de m saltos entre i y j que mayor valor de confianza indirecta proporciona.

La novedad y el punto fuerte del modelo propuesto en [7] es que todos los factores que influyen en la confianza se representan como pesos dinámicos que se adaptan con el tiempo según la política de confianza de cada nodo, gracias a ciertas realimentaciones. Esta forma de representación simplifica enormemente tanto el modelado, como el cálculo en sí de la confianza.

Los autores de *TPOD* [8] proponen incentivar a aquellos nodos de una red P2P de difusión de contenidos que más tiempo permanezcan en ella y que mayor cantidad de información difundan, pues se considera que son los más fiables y con mayor reputación de la red.

GroupRep [9] es un modelo donde las relaciones de confianza se clasifican en tres niveles: las relaciones de confianza entre grupos, entre grupos y pares y solamente entre pares.

En el caso de *Patrol-F* [10] el modelo incorpora varios conceptos importantes en el cálculo de la reputación de un par como son: experiencias directas y valores de reputación, la credibilidad de un nodo para dar recomendaciones, la devaluación de la información con el tiempo basada en un factor dinámico de devaluación, las primeras impresiones y la jerarquía del sistema de nodos. Para ello utiliza lógica difusa.

Por último, TDTM [11] es el único modelo que hemos encontrado en el que se aplique el sistema de colonia de hormigas. En él se identifica la feromona con la confianza y la heurística con la distancia entre dos nodos. Se define un modelo en el que la confianza aumenta cuando dos nodos cualquiera interactúan y disminuye con el tiempo si no hay ninguna interacción. También define un mecanismo para modificar la feromona del camino que conecta a dos nodos que han interactuado. Este modelo además presupone la existencia de una PKI.

El modelo que nosotros proponemos también aplica el sistema de colonia de hormigas, pero no presupone la existencia de una PKI y además tiene en cuenta la posibilidad de que las interacciones no sean satisfactorias.

3. Cuerpo del trabajo

3.1. Sistema de Colonia de Hormigas

La Optimización basada en Colonia de Hormigas [12] (en adelante OCH) es una metaheurística bioinspirada que permite diseñar algoritmos sencillos de entender, rápidos y con buen rendimiento, para problemas de optimización que se puedan representar en forma de grafo con pesos.

Así, la hormiga artificial es un agente que recuerda los nodos que ha recorrido, utilizando para ello una lista de nodos visitados L. Al finalizar, esta lista contiene la solución construida por la hormiga. Además, en cada paso, estando en el nodo c elige hacia qué nodo s moverse de entre los vecinos de c que no hayan sido visitados aún ($J(c) = \{s \mid \exists\ a_{cs} \wedge s \notin L\}$), según una regla probabilística de transición y dejando en dicho arco a_{cs} un cierto rastro de feromona τ_{cs}.

Por tanto, la probabilidad de que la hormiga k situada en el nodo c se desplace al nodo s viene dada por la siguiente regla de transición:

$$p_k(c,s) = \begin{cases} \frac{[\tau_{cs}]^\alpha \cdot [\eta_{cs}]^\beta}{\sum_{u \in J_k(c)} [\tau_{cu}]^\alpha \cdot [\eta_{cu}]^\beta} & \text{si } s \in J_k(c) \\ 0 & \text{en otro caso} \end{cases} \quad (1)$$

donde τ_{cs} es la feromona del arco a_{cs}, η_{cs} es la información heurística del arco a_{cs}, $J_k(c)$ es el conjunto de nodos alcanzables desde c no visitados aún por la hormiga k y α y β son pesos que establecen un equilibrio entre la importancia de la información memorística y la heurística, respectivamente.

Concretamente, el Sistema de Colonia de Hormigas [13] (SCH) incluye una regla de transición proporcional pseudo-aleatoria con la siguiente forma:

$$s = \begin{cases} \arg\max_{u \in J_k(c)} [\tau_{cu}]^\alpha \cdot [\eta_{cu}]^\beta & \text{si } q \leq q_0 \\ S & \text{en otro caso} \end{cases} \quad (2)$$

donde s es el nodo escogido por la hormiga k en su siguiente movimiento, q es un uniforme en [0,1], q_0 es la probabilidad con la que se escoge determinísticamente el arco más prometedor y S es un nodo aleatorio seleccionado según la regla de transición de la OCH.

El algoritmo de Sistema de Colonia de Hormigas tiene el siguiente aspecto:

algoritmo SCH **devuelve** mejor solución encontrada y su coste;
  Inicialización de parámetros;
  Para $It = 1$ **hasta** $Numero_de_Iteraciones$ **hacer**;
    Para $k = 1$ **hasta** $Numero_de_Hormigas$ **hacer**;
      $L[k][1] \leftarrow$ nodo inicial;
    /* Construcción de soluciones por las hormigas */;
    Para $i = 2$ **hasta** $Numero_de_Nodos$ **hacer**;
      Para $k = 1$ **hasta** $Numero_de_Hormigas$ **hacer**;
        $L[k][i] \leftarrow$ Regla_transición$(L[k], \tau, \eta)$;
        Actualización_local_feromona$(L[k][i-1], L[k][i])$;
    Para $k = 1$ **hasta** $Numero_de_Hormigas$ **hacer**;
      $Coste[k] \leftarrow C(L[k])$;
      $Mejor_Actual \leftarrow Mejor(L[k])$;
    Si $C(Mejor_Actual)$ es mejor que $C(Mejor_Global)$ **entonces**;
      $Mejor_Global \leftarrow Mejor_Actual$;
    Para $i = 1$ **hasta** $Numero_de_Nodos$ **hacer**;
      Actualización_global_feromona$(Mejor_Global[k][i], Mejor_Global[k][i+1], C(Mejor_Global))$;
  devolver $Mejor_Global$;

Algoritmo 1: Sistema de Colonia de Hormigas (SCH)

Ya se ha visto cuál es la regla de transición del SCH. A continuación se muestra la actualización local de feromona:

$$\tau_{cs}(t) = (1 - \varphi) \cdot \tau_{cs}(t-1) + \varphi \cdot \tau_0 \tag{3}$$

Con esta operación, la feromona asociada a un arco disminuye cada vez que lo visita una hormiga y así los arcos ya visitados van siendo menos prometedores según los recorren más hormigas en la iteración actual, lo que favorece la exploración de arcos no visitados. Se consigue de esta forma que las hormigas tiendan a no converger a soluciones parecidas en la iteración actual

Por su parte, la actualización global de feromona sólo se aplica sobre aquellos arcos que constituyen la mejor solución global y depende de la calidad de ésta última. Tiene la siguiente forma:

$$\tau_{cs}(t) = (1 - \rho) \cdot \tau_{cs}(t-1) + \rho \cdot \frac{1}{C(S_{Mejor_Global})} \tag{4}$$

3.2. TACS. Trust Ant Colony System

Este modelo se plantea de la siguiente manera: un nodo cualquiera de la red P2P solicita un servicio concreto a los demás nodos de la red. En ese momento, el nodo solicitante actúa como cliente de un servicio que ofrecen algunos (o todos) de los demás nodos de la red, que entonces actúan como servidores. El modelo busca aquel camino "óptimo" que permita a un cliente dado alcanzar un determinado servicio ofrecido por otro nodo de la red que no tiene por qué estar directamente accesible.

En general, para poder aplicar la OCH a un problema es necesario que éste pueda ser representado como un grafo con pesos. Esto se ajusta perfectamente al escenario con el que nos encontramos ya que una red P2P puede ser representada instantáneamente como un grafo.

En cuanto a los pesos de las aristas del grafo, en este modelo se ha decidido identificar la feromona de la arista a_{cs}, τ_{cs}, con la confianza que el nodo c tiene en el nodo s, $T_{c \to s}$. Sin embargo, cabe destacar que existen distintos rastros de feromona para cada servicio que se solicite, pues no sería justo que a un nodo que es muy bueno ofreciendo un servicio, por ejemplo S_1, se le penalice porque también ofrece un servicio, pongamos S_2, pero en muy malas condiciones. Haciendo esta distinción, cada nodo tiene unos niveles de confianza en el resto de nodos de la red P2P, para cada servicio que solicite.

Por otra parte, la heurística del arco a_{cs}, η_{cs}, se ha identificado con la similitud entre el servicio solicitado por el nodo c y el ofrecido por el nodo s (que puede ser distinto o no del servicio finalmente entregado, dependiendo de la bondad de s). Esta distinción se entenderá mejor más adelante.

También es importante destacar que tanto la feromona τ_{cs}, como la heurística η_{cs} de un arco a_{cs} toman valores en el intervalo $[0, 1]$ (aunque en realidad se han definido unos umbrales que hacen que concretamente $\tau_{cs} \in [0'01, 0'99]$, para evitar inconsistencias matemáticas al multiplicar por 0).

En la figura 1 se muestran dos escenarios en los que se tiene un nodo origen C, que actúa como cliente que solicita un servicio y varios nodos S que actúan como servidores, ofreciendo dicho servicio (aunque en el caso (b) en concreto algunos servidores podrían actuar simplemente como intermediarios).

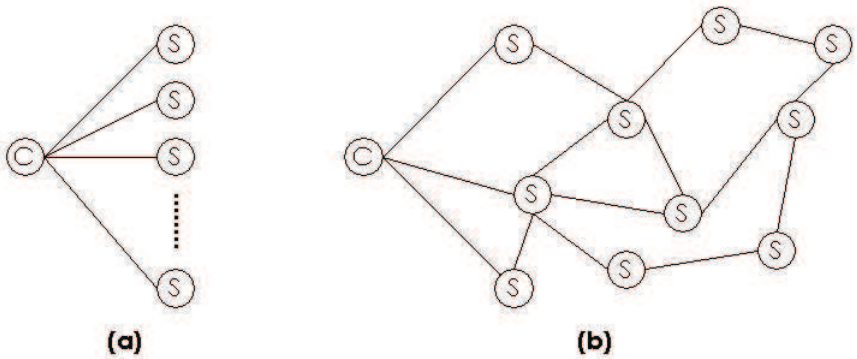

Figura 1. Topologías de redes P2P

De este modo, las hormigas parten del nodo origen C y buscan aquel nodo destino S que sea "óptimo" ofreciendo el servicio requerido. Si un nodo S no ofrece el servicio solicitado, jamás será visitado por una hormiga ya que el valor de su heurística será $\eta_S = 0$ y por tanto tendrá un probabilidad de 0.0 de ser escogido según la regla de transición del SCH (fórmulas (1) y (2)).

Igualmente, si un nodo se desconecta de la red o se conecta uno nuevo, el propio SCH se reajusta automática y dinámicamente para encontrar nuevas rutas que conduzcan hacia el servidor "óptimo".

Una vez el SCH ha finalizado, el cliente interactúa con aquel servidor que el SCH en cuestión haya seleccionado como mejor. Una vez realizada la interacción, se evalúa la satisfacción del cliente con el servicio recibido. Si el cliente está satisfecho, todo queda como está, pero si el cliente no está satisfecho del todo con el servicio recibido, entonces se produce una penalización en la confianza otorgada al servidor con el que se ha interactuado. Dicha penalización viene en forma de evaporación adicional de la feromona de aquellas aristas que componen el camino desde el nodo C hasta el servidor seleccionado.

Por último, es importante destacar la diferencia entre $\tau_{cs}(0)$ y τ_0. Por un lado, $\tau_0 \sim [0'45, 0'55]$ es la feromona inicial de todos los arcos de la red, antes de realizar ninguna interacción. $\tau_{cs}(0)$, sin embargo, representa el rastro del arco a_{cs} cuando el cliente C lanza el algoritmo de colonia de hormigas para buscar el mejor servidor con el que interactuar. Dicho valor $\tau_{cs}(0)$, por tanto, irá cambiando conforme el cliente vaya solicitando más servicios y conforme se comporten los servidores seleccionados para ofrecer dichos servicios. Esto es, cambia con el número de veces que se ejecute el algoritmo.

En resumen, las etapas que componen este modelo son las siguientes, tal y como se muestran en la figura 2:

1. El cliente C ejecuta TACS para buscar el servidor S que sea "óptimo" ofreciendo el servicio s.
2. TACS ejecuta el algoritmo SCH y las hormigas van modificando los rastros de feromona.
3. TACS termina de ejecutarse, habiendo seleccionado el camino "óptimo" al servidor S'.
4. TACS informa al cliente C de que el servidor "óptimo" encontrado es S'.
5. El cliente C solicita al servidor S' el servicio s.
6. El servidor S' proporciona al cliente C el servicio s'.
7. El cliente C evalúa su satisfacción con el servicio s' recibido.
8. Si el cliente C no está satisfecho con el servicio s' recibido, penaliza al servidor S' evaporando feromona en los arcos que conducen desde C hasta S'.

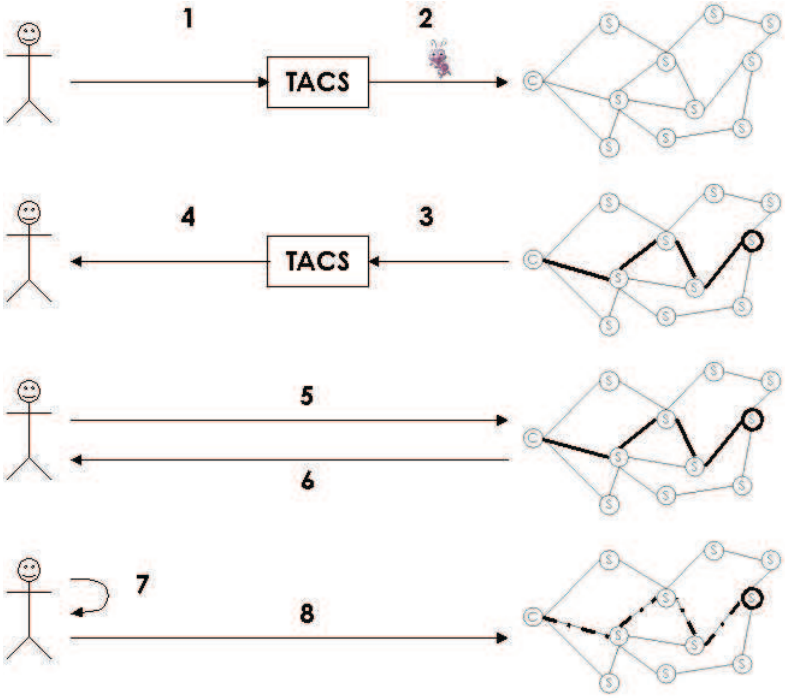

Figura 2. Etapas del modelo TACS

3.3. Características específicas del modelo

Una vez se ha descrito el comportamiento general del modelo, a continuación se describirán las particularidades del mismo. Se explicará con detalle cómo se realiza la actualización de feromona, tanto local como global, cómo se calcula el coste de una solución y cómo se lleva a cabo una penalización en caso de que el cliente no esté satisfecho con el servicio recibido.

Antes que nada cabe decir que de los dos escenarios planteados en la figura 1, por el momento nos centraremos en el (a), consistente en un cliente con N servidores directos, sin intermediarios. El escenario (b), no obstante, queda planteado como trabajo futuro.

En primer lugar hablaremos de la actualización de la feromona. Recordemos que tanto la actualización local, como la global, tenían la siguiente forma:

$$\tau_{cs}(t) = (1-\lambda)\cdot\tau_{cs}(t-1) + \lambda\cdot z \tag{5}$$

siendo $z = \tau_0$ en el caso de la actualización local y $z = \frac{1}{C(S_{Mejor_Global})}$, en el caso de la global.

De aquí se puede obtener lo siguiente:

$$\begin{aligned}
&\tau_{cs}(1) = (1-\lambda)\cdot\tau_{cs}(0) + \lambda\cdot z\\
&\tau_{cs}(2) = (1-\lambda)^2\cdot\tau_{cs}(0) + (1-\lambda)\cdot\lambda\cdot z + \lambda\cdot z\\
&\tau_{cs}(3) = (1-\lambda)^3\cdot\tau_{cs}(0) + (1-\lambda)^2\cdot\lambda\cdot z + (1-\lambda)\cdot\lambda\cdot z + \lambda\cdot z\\
&\vdots\\
&\tau_{cs}(t) = (1-\lambda)^t\cdot\tau_{cs}(0) + \lambda\cdot z\sum_{i=0}^{t-1}(1-\lambda)^i
\end{aligned}$$

Y puesto que $\lambda \in [0,1]$ se obtiene finalmente que

$$\lim_{t\to\infty}\tau_{cs}(t) = \lim_{t\to\infty}(1-\lambda)^t\cdot\tau_{cs}(0) + \lim_{t\to\infty}\lambda\cdot z\cdot\frac{1-(1-\lambda)^t}{\lambda} = 0\cdot\tau_{cs}(0) + \lambda\cdot z\cdot\frac{1}{\lambda} = z \tag{6}$$

Es decir, cuando el número de iteraciones tiende a infinito y se usa la actualización de feromona que se muestra en la fórmula (5), entonces el rastro de feromona tiende a z.

Una vez conocido esto, se comprobó que para el caso de la actualización local de feromona el valor de z no podía ser ni $\tau_{cs}(0)$, ni τ_0, como se propone en la fórmula (3) original del algoritmo 1. Una evaporación local de feromona, por pequeña que fuera, provocaba que los valores medios de feromona no convergieran adecuadamente hacia los valores esperados, como se verá en el siguiente apartado de experimentos y resultados.

Así, se decidió que la actualización local de feromona no fuera una evaporación de ésta, sino un aporte de la misma. Por lo tanto, se propuso que el valor de z en este caso fuera el siguiente:

$$z_1 = \tau_{cs}(0)\cdot(1 + (1-\tau_{cs}(0))\cdot\eta_{cs}\cdot\varphi) \tag{7}$$

Por lo que la actualización local de feromona en TACS es

$$\tau_{cs}(t) = (1-\varphi)\cdot\tau_{cs}(t-1) + \varphi\cdot z_1 \tag{8}$$

Sabiendo que $\tau_{cs}(0), \eta_{cs}, \varphi \in [0,1]$, el incremento nunca superará el 100 %. Aún así, después de cada aporte de feromona, se comprueba que ésta se halle dentro de los umbrales establecidos. Con el factor $1-\tau_{cs}(0)$ se consigue que cuando $\tau_{cs}(0) \to 0$, el aporte de feromona sea mayor, permitiendo recuperar la confianza más rápidamente. Si por contra $\tau_{cs}(0) \to 1$, entonces se aporta menos cantidad de feromona.

En cuanto a la actualización global de feromona, puesto que ésta se lleva a cabo en los arcos que constituyen en la mejor solución hasta el momento, se debe llevar a cabo un aporte que sea superior al que se realiza en la actualización local. Bajo esta premisa, el valor de z en este caso es el siguiente:

$$z_2 = \tau_{cs}(0)\cdot(1 + \tau_{cs}(0)\cdot\eta_{cs}\cdot C(S_{Mejor_Global})) \tag{9}$$

Teniéndose la siguiente actualización global de feromona para TACS:

$$\tau_{cs}(t) = (1-\rho)\cdot\tau_{cs}(t-1) + \rho\cdot z_2 \tag{10}$$

Como veremos a continuación $C(S_{Mejor_Global}) \in [0,1]$ y empíricamente se comprueba también que $C(S_{Mejor_Global}) > \varphi$, por lo que se tiene que $z_2 > z_1$, lo que significa que el aporte de feromona en la actualización global es mayor que el que tiene lugar en la actualización local.

Para calcular la calidad de una solución S_k, $C(S_k)$, se han tenido en cuenta dos factores: por un lado el porcentaje del total de hormigas que han elegido esa misma solución y por otra parte la longitud de dicha solución. Partiendo de la base de que un servicio ofrecido con muchos intermediarios de por medio es menos "fiable" que otro ofrecido con pocos intermediarios o incluso ninguno, interactuando directamente con el servidor final, se debe dar más calidad a aquellas soluciones cuya longitud sea menor. De esta forma se decidió que la calidad de una solución S_k se calculara de la siguiente forma:

$$C(S_k) = \frac{\%H_k}{\sqrt{Longitud(S_k)}} \tag{11}$$

Sin embargo, puesto que, como ya hemos comentado, en este primer trabajo nos hemos centrado en el escenario en que no hay intermediarios (figura 1.(a)), la expresión anterior se simplifica, quedando como

$$C(S_k) = \%H_k \tag{12}$$

Definiendo la calidad de una solución como se indica en (11), cabría preguntarse bajo qué condiciones sería más interesante una solución con intermediarios que otra en la que hubiera una comunicación directa. En definitiva cabe preguntarse cuándo ocurre que $C(S_1) < C(S_2)$ si $Longitud(S_1) = 1 < Longitud(S_2)$.

$$C(S_1) < C(S_2) \qquad \%H_1 < \frac{\%H_2}{\sqrt{Longitud(S_2)}}$$

$$\frac{\%H_1}{\sqrt{Longitud(S_1)}} < \frac{\%H_2}{\sqrt{Longitud(S_2)}} \qquad \%H_1 \cdot \sqrt{Longitud(S_2)} < \%H_2$$

Con todo lo visto hasta ahora podemos describir el algoritmo TACS, que no es sino la adaptación del algoritmo 1, siguiendo las modificaciones antes mencionadas.

algoritmo TACS **devuelve** mejor solución encontrada y su coste;
Para $It = 1$ **hasta** $Numero_de_Iteraciones$ **hacer**;
  Para $k = 1$ **hasta** $Numero_de_Hormigas$ **hacer**;
    $L[k][1] \leftarrow$ nodo C;
  /* Construcción de soluciones por las hormigas */;
  Para $k = 1$ **hasta** $Numero_de_Hormigas$ **hacer**;
    $s \leftarrow$ Regla_transición$(L[k], \tau, \eta)$;
    $L[k][i] \leftarrow s$;
    $\tau_{cs}(It) = (1-\varphi) \cdot \tau_{cs}(It-1) + \varphi \cdot \tau_{cs}(0) \cdot (1 + (1 - \tau_{cs}(0)) \cdot \eta_{cs} \cdot \varphi)$;
  Para $k = 1$ **hasta** $Numero_de_Hormigas$ **hacer**;
    $Calidad[k] \leftarrow \%H_k$;
    $Mejor_Actual \leftarrow Mejor(L[k])$;
  Si $C(Mejor_Actual)$ es mejor que $C(Mejor_Global)$ **entonces**;
    $Mejor_Global \leftarrow Mejor_Actual$;
  Para $i = 1$ **hasta** $Numero_de_Nodos$ **hacer**;
    $\tau_{cs}(It) = (1-\rho) \cdot \tau_{cs}(It-1) + \rho \cdot \tau_{cs}(0) \cdot (1 + \tau_{cs}(0) \cdot \eta_{cs} \cdot C(S_{Mejor_Global}))$;
devolver $Mejor_Global$;

Algoritmo 2: Trust Ant Colony System (TACS)

Como se puede observar, ya no hay inicialización de los rastros de feromona, pues esto sólo se hace al principio de todo, antes de que comiencen todas las interacciones. También se aprecia que en la construcción de las soluciones por parte de las hormigas se ha eliminado el bucle sobre el número total de nodos, pues en nuestro escenario sin intermediarios todas las soluciones tendrán longitud 1.

Por otra parte, los servicios ofrecidos por los nodos de la red se han definido de forma genérica con tan sólo cuatro parámetros: un nombre o identificador del servicio, su precio, su calidad y su tiempo de entrega. Sin embargo, esta definición podría expresarse de una forma aún más genérica. Para ello se tendría que definir un servicio S como un conjunto de atributos medibles $x_i \in \mathbb{R}$.

Además, en el cliente se definen tres factores que son los pesos que se dan a cada característica de los servicios. Es decir, se tiene el peso del precio del servicio, el peso de la calidad y el del tiempo de entrega. De forma genérica se podría tener tantos pesos γ_i como atributos tuviera el servicio.

Así, si se tienen dos servicios $S_1 = (x_1^1, x_2^1, \ldots, x_n^1)$ y $S_2 = (x_1^2, x_2^2, \ldots, x_n^2)$, la similitud entre ellos se calcula de la siguiente forma:

$$Sim(S_1, S_2) = \frac{\text{mín}\left(1, \frac{x_1^1}{x_1^2}\right) \cdot \gamma_1 + \text{mín}\left(1, \frac{x_2^1}{x_2^2}\right) \cdot \gamma_2 + \cdots + \text{mín}\left(1, \frac{x_n^1}{x_n^2}\right) \cdot \gamma_n}{\sum_{i=1}^{n} \gamma_i} \tag{13}$$

En realidad se tendrá $\frac{x_i^1}{x_i^2}$ o $\frac{x_i^2}{x_i^1}$ dependiendo de si para el cliente es conveniente que crezca o decrezca x_i. Por ejemplo, al cliente le interesa que la calidad del servicio sea alta, mientras que el precio sea el menor posible.

Esta definición nos sirve para identificar la satisfacción de un cliente con la similitud entre el servicio solicitado por dicho cliente y el servicio realmente proporcionado por el servidor escogido. Es decir, el cliente estará tanto más satisfecho cuanto más se parezca (o incluso mejore) el servicio recibido al servicio inicialmente solicitado.

$$Sat(C) = Sim(S_i, S_i') \in [0, 1] \tag{14}$$

Como se vio en la fórmula (8), la actualización local de feromona no es una evaporación, como se propone originalmente en el SCH, sino que es un aporte de feromona. Con esto se consiguió que los valores de feromona convergieran adecuadamente, creciendo, cuando todos los servidores son totalmente confiables y honestos.

Sin embargo, cuando un cliente se sentía defraudado por el servicio obtenido de un servidor, a pesar de eso, su rastro de feromona (es decir, su confianza en ese servidor) había crecido mucho. Se hizo necesario, por tanto, realizar una penalización en forma de evaporación de feromona en aquellas interacciones entre cliente y servidor que no fueran del todo satisfactorias para el primero.

Así se decidió ligar la penalización de un servidor con la satisfacción que un cliente hubiera tenido con el servicio proporcionado por dicho servidor. Sabiendo que $Sat(C) \in [0, 1]$, si $Sat(C) \geq 0'5$ se entiende que el cliente está satisfecho con el servicio recibido, en mayor o menor medida. Por su parte, si $Sat(C) < 0'5$ entonces el cliente no está satisfecho con el servicio recibido.

Se optó por dar un tratamiento diferente a estas dos situaciones. Por consiguiente, si $Sat(C) \geq 0'5$, es decir, si el cliente sí está satisfecho con el servicio recibido, entonces se tiene la siguiente penalización:

$$\tau_{cs} = \tau_{cs} - \varphi \cdot 2 \cdot (1 - Sat(C)) \tag{15}$$

Como $Sat(C) \in [0'5, 1]$, entonces $(1 - Sat(C)) \in [0, 0'5]$ y $2 \cdot (1 - Sat(C)) \in [0, 1]$.

Por otra parte, si $Sat(C) < 0'5$ entonces la penalización tiene el siguiente aspecto:

$$\tau_{cs} = (\tau_{cs} - \varphi) \cdot Sat(C) \tag{16}$$

Esta penalización para el caso en que el cliente no esté satisfecho es bastante fuerte, pues supone multiplicar por un factor inferior a 0'5 el rastro actual de feromona. Poniéndonos en el extremo, si el rastro actual de feromona fuera $\tau_{cs} = 0'99$ (esto es, una confianza prácticamente total en dicho servidor), pero sin embargo la satisfacción del cliente fuera $Sat(C) = 0'01$, entonces el rastro de feromona tras la penalización sería $\tau_{cs} \approx 0'99 \cdot 0'01 = 0'0099 \simeq 0'01$. Es decir, pasaría de un extremo de confianza al otro totalmente opuesto (si bien es cierto, que la traición de la confianza también ha sido máxima).

3.4. Experimentación realizada

En este apartado presentaremos los resultados obtenidos a partir de todos los experimentos realizados. Veremos cómo los niveles medios de feromona convergen adecuadamente hacia los valores esperados. Para llevar a cabo una experimentación lo más amplia posible y que abarcara el mayor número de situaciones que pudieran plantearse, se decidió realizar las pruebas bajo distintos escenarios. Éstos son:

1. Todos los servidores ofrecen exactamente el mismo servicio que el que solicita el cliente (es decir, $\eta_{cs} = 1 \; \forall \; s$). Y además tienen una bondad máxima, esto es, cuando el cliente interactúa con cualquiera de ellos, todos le proporcionan el servicio que realmente le habían ofrecido (teniendo el cliente una satisfacción máxima).
2. De nuevo $\eta_{cs} = 1 \; \forall \; s$, pero esta vez la bondad de todos los servidores es mínima, lo que significa que cada vez que el cliente interactúa con cualquiera de ellos recibe un servicio totalmente insatisfactorio para él.
3. $\eta_{cs} = 1 \; \forall \; s$, también, pero ahora la mitad de los servidores tienen una bondad máxima y la otra mitad una bondad mínima.
4. Los servidores tienen una bondad oscilante, es decir, durante un periodo son muy buenos y durante el siguiente son muy malos, y así sucesivamente.

Los valores utilizados para los distintos parámetros del algoritmo son los siguientes:

- $\tau_0 \sim [0'45, 0'55]$
- $q_0 = 0'98$
- $\varphi = 0'1$
- $\rho = 0'1$
- $\alpha = 1'0$
- $\beta = 1'0$
- $N_I = \lceil 10 \cdot \ln(N_S) \rceil$
- $N_H = \frac{5}{4} \cdot N_S$

Siendo N_I el número de iteraciones, N_H el número de hormigas y N_S el número de servidores directos. Todas las experimentaciones se realizarán para $N_S \in \{10, 20, 30, 40, 50\}$ ya que consideramos que un escenario en que un cliente P2P interactúa directamente con 50 servidores es suficiente. Además, como se verá en el análisis de los resultados, con estos valores nos basta para poder obtener las conclusiones deseadas. Se han probado muchas combinaciones de valores para todos los parámetros, pero los que mejores resultados han arrojado son los que arriba se muestran.

Para cada escenario se presentarán dos gráficas. La primera de ellas mostrará la evolución de la feromona media de todos los arcos de la red, según se vaya ejecutando el algoritmo, es decir

$$\tau^M = \frac{\sum_{i=1}^{N_S} \tau_{cs_i}}{N_S} \tag{17}$$

La segunda gráfica no es menos importante que la primera. En ella se representa la desviación típica del número de veces que ha sido seleccionado cada servidor para ofrecer un servicio. En un escenario en el que todos los servidores tienen la misma bondad, en teoría y probabilísticamente hablando, todos los servidores deben ser elegidos aproximadamente el mismo número de veces. No sería válido que aunque los valores medios de feromona convergieran a los valores deseados, sólo unos pocos servidores fueran siempre elegidos y el resto nunca tuvieran oportunidad de interactuar con el cliente.

Sabemos que la fórmula general de la desviación típica es

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}}$$

Si para nosotros x_i representa el número de veces que el servidor i-ésimo ha sido elegido para interactuar con el cliente, del total de N_E veces que se ha ejecutado el algoritmo, entonces la fórmula para calcular la desviación típica que nosotros buscamos es

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N_S}(\frac{x_i}{N_E} - \frac{1}{N_S})^2}{N_S}} \tag{18}$$

Así, un valor alto de σ indica que no se han elegido por igual a todos los servidores. Del mismo modo, si $\sigma \to 0$ eso quiere decir que todos los servidores han sido seleccionados equitativamente, el mismo número de veces, que es lo que nosotros iremos buscando.

Visto todo esto, en el primer escenario en el que todos los servidores tienen la máxima bondad, el resultado esperado es que todos los rastros de feromona converjan a su valor máximo, lo más rápido posible, a la vez que todos los servidores, por ser igual de buenos, sean elegidos, en término medio, el $\frac{100}{N_S}$ % de las veces.

Por ejemplo, si $N_S = 10$, se espera que cada servidor sea elegido el $\frac{100}{10}\,\% = 10\,\%$ del total de las N_E veces que se ejecuta el algoritmo. Y tras la primera ejecución del algoritmo la desviación típica sería la siguiente

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N_S}(\frac{x_i}{N_E} - \frac{1}{N_S})^2}{N_S}} = \sqrt{\frac{\sum_{i=1}^{10}(x_i - \frac{1}{10})^2}{10}} = \sqrt{\frac{(1-\frac{1}{10})^2 + \sum_{i=2}^{10}(0-\frac{1}{10})^2}{10}} =$$

$$= \sqrt{\frac{(0'9)^2 + 9\cdot(-0'1)^2}{10}} = \sqrt{\frac{0'81 + 0'09}{10}} = \sqrt{\frac{0'9}{10}} = \sqrt{0'09} = 0'3$$

En la figura 3 se muestran los resultados del escenario 1.

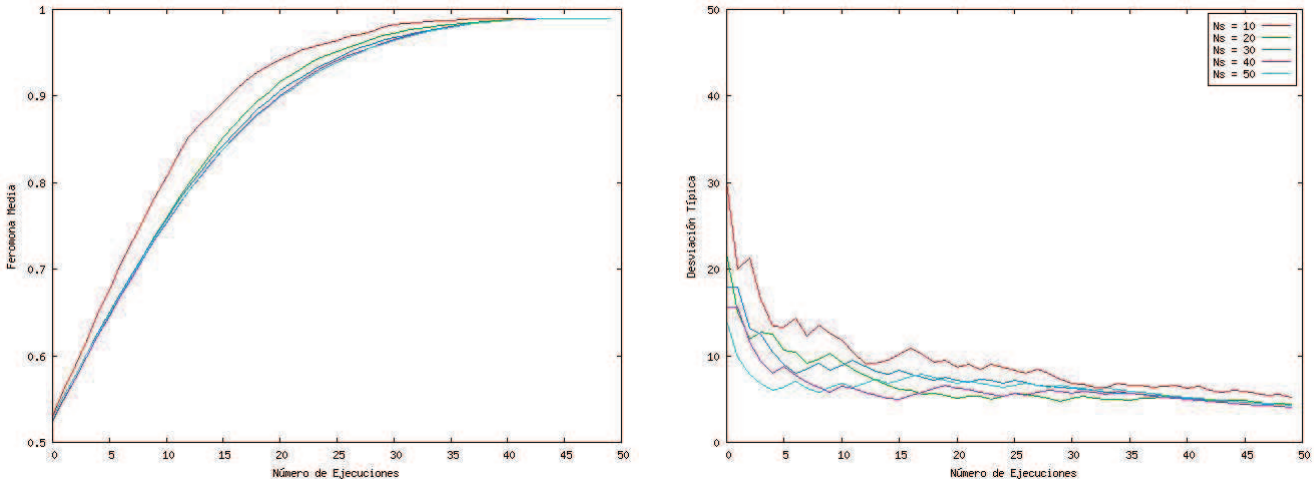

Figura 3. Escenario 1. $\eta_{cs} = 1\ \forall\ s$ y $bondad_s = 1\ \forall\ s$

En la figura de la izquierda se puede comprobar cómo el aumento de feromona es prácticamente independiente del número de servidores de la red P2P, puesto que las cinco gráficas correspondientes a $N_S = 10$, 20, 30, 40 y 50 están prácticamente superpuestas. Además también se observa que cuando se han realizado 20 ejecuciones o más, los valores medios de feromona ya superan el 90 %. Se aprecia a simple vista que el comportamiento es el esperado, pues los valores medios de feromona convergen rápidamente a su valor máximo.

Por otra parte, la gráfica de la derecha demuestra que la desviación típica toma un valor alto al principio (lo que es normal siempre) y conforme se ejecuta más veces el algoritmo, este valor va decreciendo hasta estabilizarse más o menos por debajo del 5 %, que es un valor bajísimo. Esto implica, como ya hemos explicado, que todos los servidores son elegidos aproximadamente el mismo número de veces, sin favorecer a ninguno en particular.

También se puede comprobar que, para el caso de $N_S = 10$ y $N_E = 1$, efectivamente $\sigma = 0'3$, como habíamos calculado antes. A partir de la segunda ejecución, depende qué servidores se vayan eligiendo para que esta desviación crezca o disminuya, pero como ya hemos comentado, en este caso disminuye correctamente.

En el segundo escenario, sin embargo, todos los servidores tienen una bondad mínima, es decir, "ofrecen" un servicio idéntico al solicitado por el cliente ($\eta_{cs} = 1$ siempre), pero a la hora de dar realmente dicho servicio, entregan uno muy distinto e insatisfactorio para el cliente. El resultado esperado, por tanto, será aquel en el que el valor medio de la feromona disminuya rápidamente hasta su valor mínimo. La desviación típica, no obstante, debe seguir siendo baja, pues aunque todos los servidores sean malos, son todos igual de malos.

La figura 4 muestra los resultados del escenario 2.

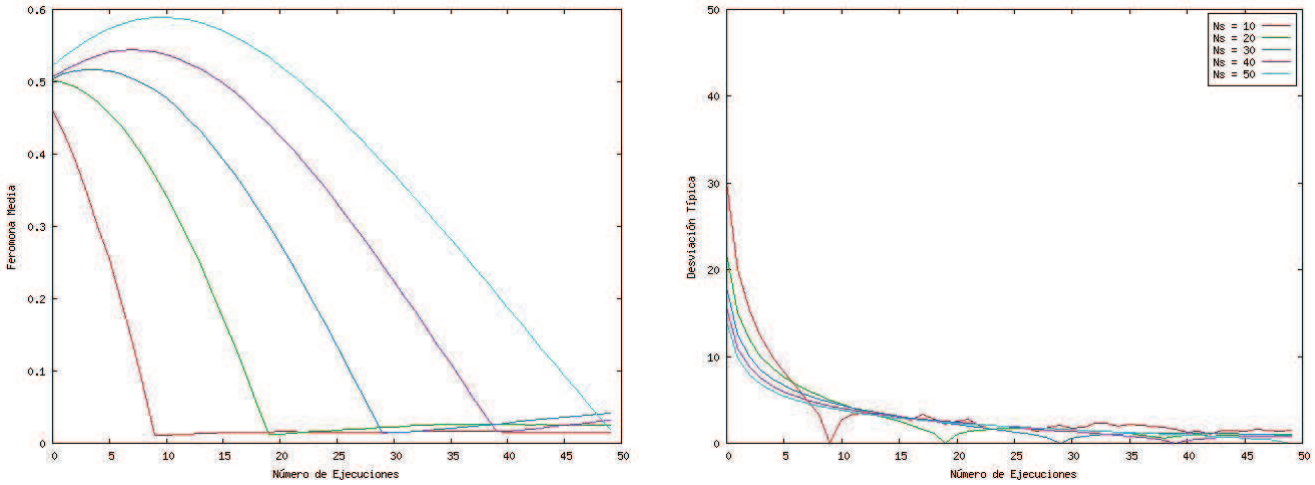

Figura 4. Escenario 2 (I). $\eta_{cs} = 1 \ \forall \ s$ y $bondad_s = 0'01 \ \forall \ s$

Como se puede observar en la gráfica de la izquierda, en este escenario el comportamiento también es el esperado: los valores medios de feromona convergen rápidamente a sus valores mínimos.

Puesto que la actualización local de feromona se definió como un aporte y no como una evaporación de feromona, la única evaporación de ésta tiene lugar en la penalización posterior a una interacción, cuando ésta última no ha sido satisfactoria para el cliente, como es el caso de este escenario.

Si observamos con atención la gráfica descubriremos que la convergencia del valor medio de feromona a su valor mínimo se alcanza cuando $N_E \rightarrow N_S^-$. Este comportamiento se explica de la siguiente manera: en cada una de las N_S primeras ejecuciones se va eligiendo un servidor distinto, y puesto que la bondad de todos ellos es mínima, se les penaliza al máximo. Así, cuando $N_E = N_S$ todos los servidores han sido seleccionados exactamente una vez, penalizados al máximo y por tanto la feromona media es mínima.

Si el N_S es alto (mayor que 30, aproximadamente) y el N_E bajo (menor que 10, aproximadamente), el valor medio de feromona crece un poco. Esto se explica porque, aunque algunos servidores ya hayan sido seleccionados y penalizados convenientemente, aún hay muchos más servidores cuya feromona se ha incrementado con la actualización local de ésta, lo que hace que en término medio la feromona aumente.

En cuanto a la desviación típica, en la gráfica de la derecha se observa cómo ésta tiene forma convexa cuando $N_E < N_S/2$, en $N_S/2$ aproximadamente tiene un punto de inflexión y es cóncava si se cumple que $N_S/2 < N_E < N_S$, alcanzando el 0 absoluto cuando $N_E \rightarrow N_S^-$. A partir de ese momento crece un poco para después decrecer lentamente hasta casi estabilizarse en valores por debajo del 2 %, lo cual son unos resultados excelentes.

La siguiente cuestión que cabría plantearse es si una vez alcanzados los valores mínimos de feromona, éstos se mantienen o los servidores son capaces de recuperarse, a pesar de seguir teniendo una bondad mínima. Pues bien, la respuesta se encuentra en la figura 5, donde se observa que efectivamente los valores medios de feromona vuelven a crecer levemente por encima del mínimo, pero acaban estabilizándose en valores que no superan el 10 % en el peor de los casos probados ($N_S = 50$).

También se demuestra que la desviación típica tiende a cero, cuanto mayor sea el N_E. Particularmente, para $N_E = 150$ ningún valor de la desviación típica supera el 0'6 %.

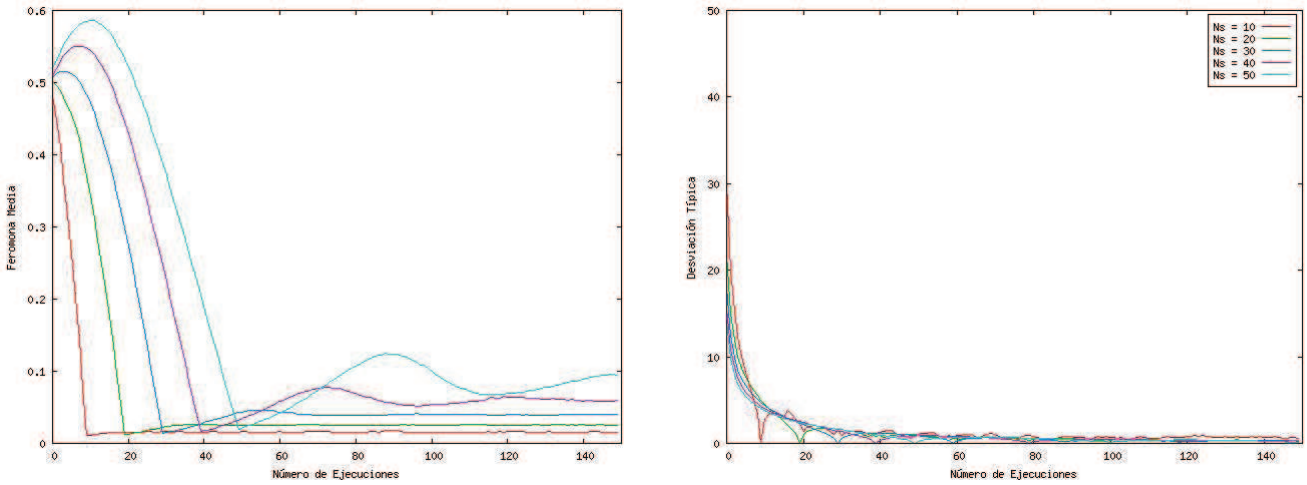

Figura 5. Escenario 2 (II). $\eta_{cs} = 1 \; \forall \; s$ y $bondad_s = 0'01 \; \forall \; s$

En el tercer escenario se tiene una red en la que la mitad de los servidores tienen una bondad máxima y la otra mitad una bondad mínima. Sin embargo, todos dicen ofrecer exactamente el mismo servicio que solicita el cliente (es decir, $\eta_{cs} = 1 \; \forall \; s$). Por lo tanto, las hormigas no pueden distinguir a priori a los servidores buenos de los malos.

No obstante, con el paso del tiempo y conforme se ejecute más veces el algoritmo, el comportamiento esperado debería ser que la feromona media de los servidores buenos convergiera a su valor máximo, mientras que la feromona media de los servidores malos debería converger a su valor mínimo.

Comenzaremos viendo primero los resultados obtenidos con la mitad de los servidores que tienen bondad máxima. Dichos resultados se muestran en la figura 6.

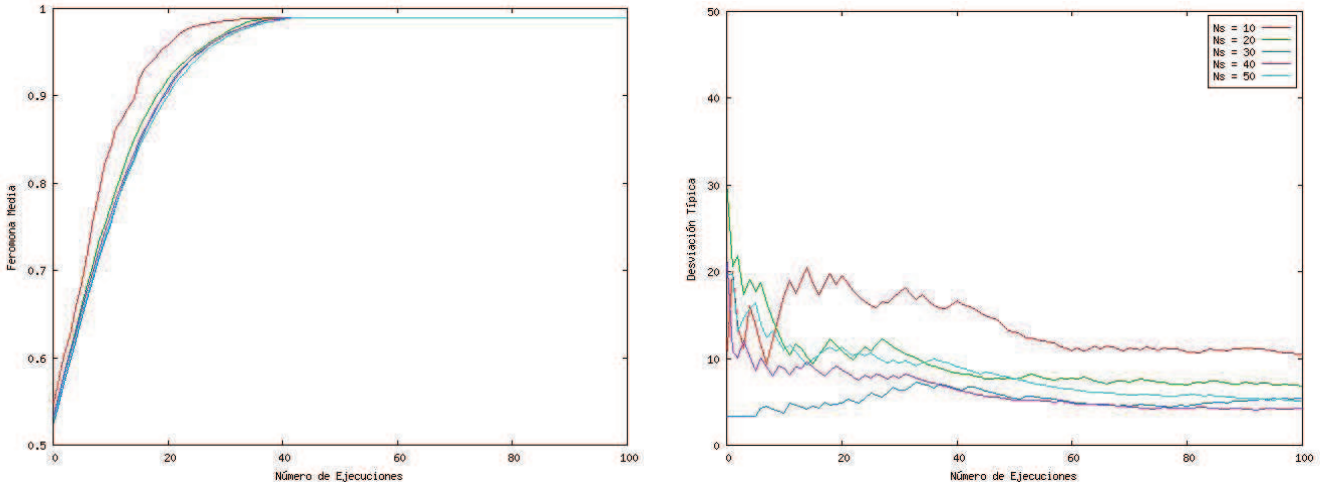

Figura 6. Escenario 3 (I). $\eta_{cs} = 1 \; \forall \; s$ y $bondad_s = 1'0$

Como bien se puede observar en la gráfica de la izquierda, el resultado obtenido es más que satisfactorio. La feromona media converge rápidamente a su valor máximo, estando dicho valor por encima del 90 % en todos los casos con 20 o más ejecuciones, aproximadamente.

Este resultado, por tanto, iguala e incluso mejora levemente al del escenario 1, pero no debemos confundirnos, ya que no podemos olvidar que ahora las gráficas representan la mitad de los servidores, es decir, tenemos 5, 10, 15, 20 y 25 servidores. Y por tanto, la convergencia es más rápida (casi el doble).

En cuanto a la desviación típica, los resultados obtenidos, aún siendo los peores obtenidos en todos los escenarios, siguen siendo buenos. En término medio, la desviación típica se encuentra por debajo del 10 % (ligeramente por encima del 10 % sólo en el caso de $N_S = 10$).

Por otro lado cabe hablar del comportamiento tenido por parte de la mitad de los servidores que tienen una bondad mínima. Como ya hemos dicho, su feromona media debería caer hasta su valor mínimo. La figura 7 muestra los resultados obtenidos en este escenario con los servidores malos.

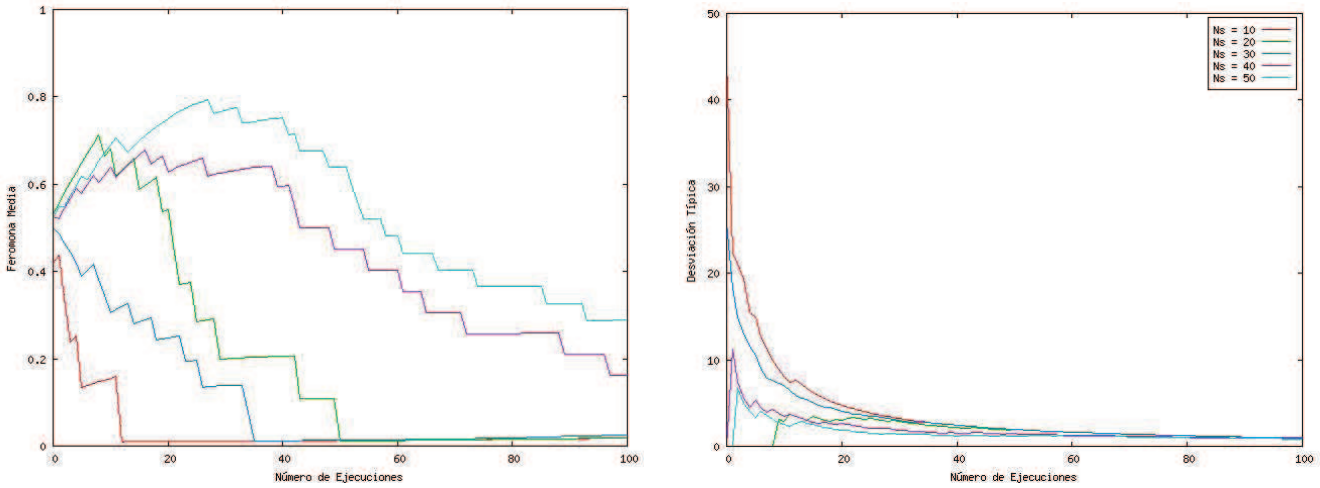

Figura 7. Escenario 3 (II). $\eta_{cs} = 1 \; \forall \; s$ y $bondad_s = 0'01$

La gráfica de la izquierda tiene la siguiente interpretación: inicialmente todos los servidores (buenos y malos) tienen aproximadamente la misma feromona y puesto que todos dicen ofrecer el servicio "perfecto", todos tienen aproximadamente la misma probabilidad de ser escogidos. Por eso la feromona media de los servidores malos crece mientras ninguno de ellos sea seleccionado para interactuar con él. Cuando se selecciona alguno de los servidores malos, tras interactuar con él y ofrecer un servicio insatisfactorio, es penalizado duramente, con lo que la feromona media disminuye.

Mientras que todas las gráficas vistas hasta el momento tendrían aproximadamente el mismo aspecto independientemente del número de veces que se crearan, ésta es distinta. Cada vez que se ejecuta el modelo puede salir una gráfica del mismo estilo, pero distinta, pues dependiendo de en qué momento sean elegidos los servidores malos (y esto es algo totalmente aleatorio), disminuirá antes o después la feromona media. En el caso óptimo, todos los $N_S/2$ servidores malos serían elegidos en las primeras N_E ejecuciones. Y entonces esta gráfica sería del mismo estilo que su equivalente del escenario 2.

Por otra parte, no podemos olvidar que mientras que estos servidores son elegidos y penalizados, disminuyendo su feromona media y con ello la probabilidad de ser elegidos, al mismo tiempo hay el mismo número de servidores en la red que son óptimos, por lo que llegará un momento en el que sólo se escogerán (probabilísticamente hablando) a los servidores buenos y nunca a los malos.

Sin embargo, esto tiene una implicación. Puesto que a partir de un cierto instante los servidores malos no son elegidos prácticamente nunca, tampoco son penalizados prácticamente nunca y como consecuencia su feromona media vuelve a crecer.

Esto, aunque pudiera parecerlo, no es un fallo, sino todo lo contrario. Cuando un servidor alcanza su valor mínimo de feromona, se le debe permitir recuperarse con cierta velocidad. Si como ocurría en el escenario 2, todos servidores que intentan recuperarse siguen teniendo una bondad mínima, todos vuelven a ser penalizados y su feromona media acaba estabilizándose, como ya vimos.

El caso del escenario 3 es un poco más complicado, puesto que los servidores malos, mientras se están recuperando, no se les penaliza hasta que alcanzan unos valores de feromona que sean lo suficientemente altos como para que, por probabilidad, puedan ser seleccionados y penalizados de nuevo. Podría incluso darse el caso (sobre todo cuanto mayor sea el N_S) que el nivel medio de feromona nunca llegara a tomar su valor mínimo, aunque se acercara bastante a él.

En la figura 8 se muestra el mismo resultado que en la figura anterior, pero con un mayor número de ejecuciones, de forma que se puede observar cómo la feromona media de los servidores malos, efectivamente vuelve a crecer hasta que toma unos valores que hacen que dichos servidores vuelvan a ser candidatos factibles a ser seleccionados y por tanto penalizados.

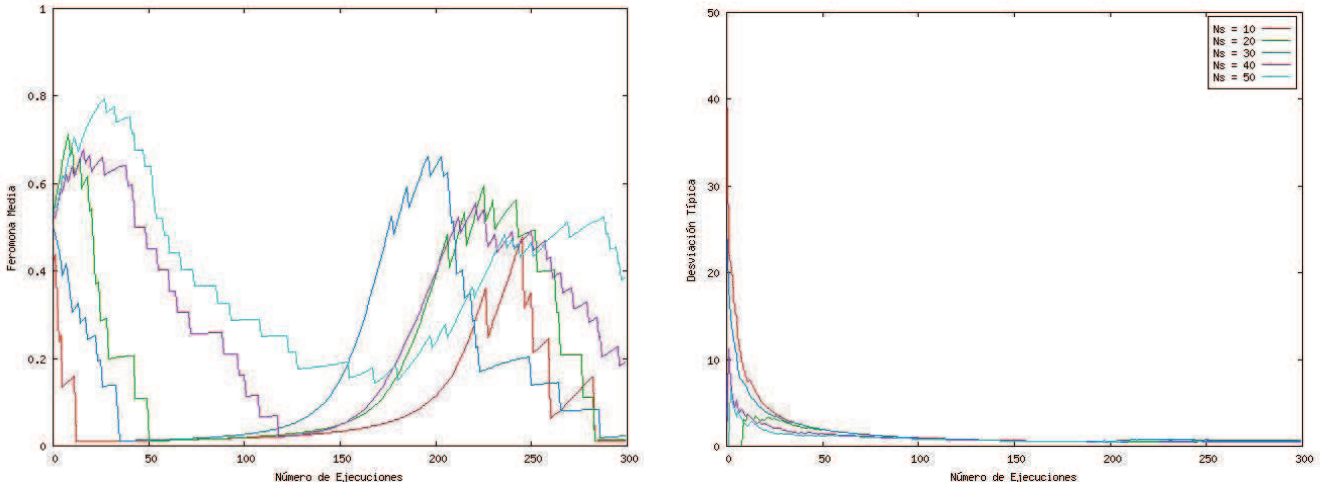

Figura 8. Escenario 3 (III). $\eta_{cs} = 1 \; \forall \; s$ y $bondad_s = 0'01$

En este escenario no hemos hablado demasiado de la desviación típica, a pesar de que, como se puede comprobar en la gráfica de la izquierda, también se obtienen resultados excelentes en este aspecto.

Para el escenario 3 en particular, pensamos que sería mucho más interesante mostrar cuántas veces eran seleccionados los servidores buenos frente a los malos. Esto nos daría una señal de cómo de bien o mal se estaría comportando nuestro modelo. Así, el comportamiento esperado sería que los servidores buenos fueran seleccionados la mayoría de las veces, mientras que los malos no lo fueran casi nunca.

Por lo tanto, decidimos construir la gráfica 9. En ella, a la izquierda se muestra el porcentaje de veces que han sido seleccionados los servidores buenos a lo largo del tiempo, mientras que la gráfica de la derecha hace lo propio con los servidores malos.

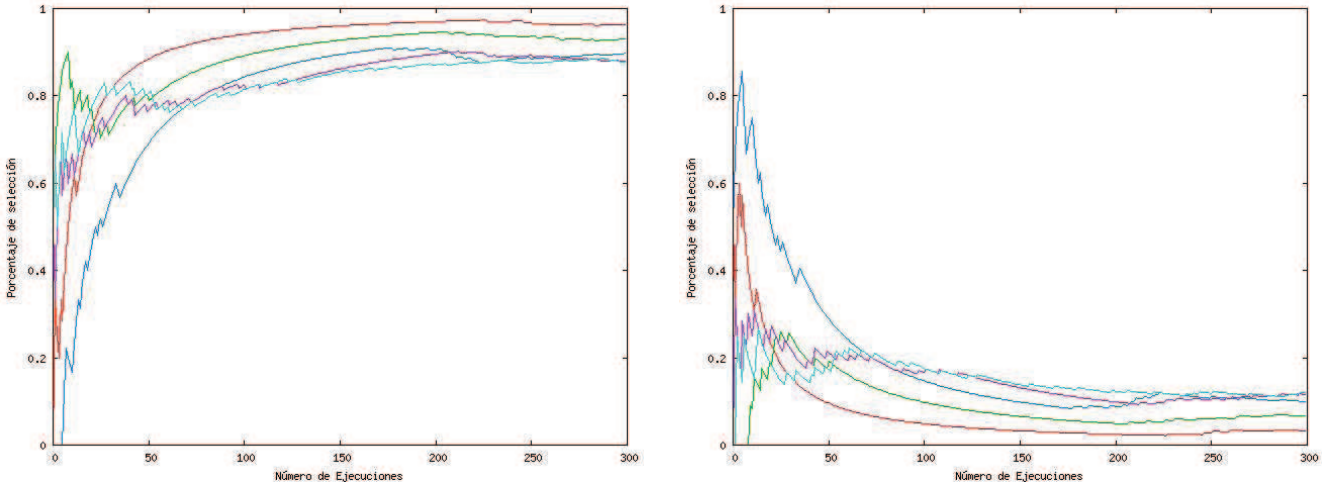

Figura 9. Escenario 3 (IV). Porcentajes de selección de los servidores buenos y malos

Como se puede comprobar, el resultado obtenido es el esperado. Con el tiempo, los servidores buenos tienden a ser elegidos en un alto porcentaje de las veces, mientras que el porcentaje de selección de los servidores malos, tiende a un valor mínimo. Inicialmente puede ocurrir cualquier cosa, puesto que por probabilidad se podrían elegir en las primeras $N_S/2$ ejecuciones justamente a todos los servidores malos, haciendo que el porcentaje de selección de éstos fuera del 100 % en esos primeros instantes.

El cuarto y último escenario es bastante interesante. Se trata de una red de servidores P2P en la que inicialmente todos tienen una bondad máxima. Cuando la feromona media de todos ellos ha alcanzado su valor máximo, entonces todos de una cambian su bondad al mínimo valor. En el momento en que el valor medio de feromona llega a su valor mínimo, entonces vuelven a modificar todos los servidores su bondad para llevarla hasta su valor máximo. Y así, este ciclo se repite indefinidamente.

La figura 10 muestra los resultados obtenidos en este escenario.

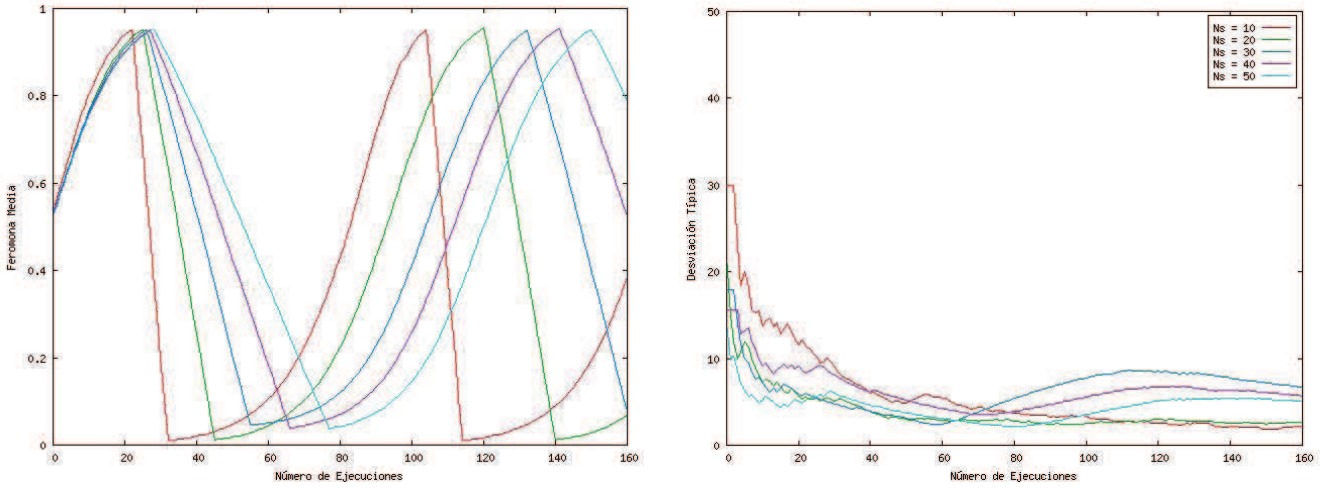

Figura 10. Escenario 4. $\eta_{cs} = 1 \; \forall \; s$ y $bondad_s$ oscilante

Estas gráficas se pueden explicar basándonos en todo lo expuesto hasta el momento. Así, se puede comprobar cómo el valor medio de la feromona en las primeras 20-30 ejecuciones supera el 95 % de su valor máximo, tal y como ocurría en el escenario 1. En ese momento, como hemos dicho, todos los servidores cambian su bondad y a partir de ese momento proporcionan servicios totalmente insatisfactorios para el cliente. Pues bien, igual que ocurría en el escenario 2, el valor medio de la feromona cae hasta el mínimo permitido en N_S ejecuciones.

Una vez alcanzado el valor mínimo para la feromona media, entonces los servidores vuelven a cambiar de actitud, teniendo a partir de ese instante una bondad máxima. Por lo tanto, el valor medio de la feromona efectivamente se recupera tan rápido como es posible, sin arriesgarse a aumentar demasiado rápido, ya que esto podría conducir a aumentar la feromona de un servidor que realmente no lo mereciera.

Si nos fijamos bien, veremos que la recuperación de la feromona desde su valor mínimo, hasta el máximo requiere de aproximadamente 75 ejecuciones en todos los casos, independientemente del N_S. Quizás en la siguiente tabla se vea más clara esta cifra.

N_S	N_{E_1}	N_{E_2}	$N_{E_2} - N_{E_1}$
10	30	105	75
20	45	120	75
30	55	130	75
40	65	140	75
50	75	150	75

Donde N_{E_1} es el número de ejecuciones aproximado en que se alcanza por primera vez el valor mínimo de feromona y N_{E_2} es el número aproximado de ejecuciones necesarias para alcanzar por segunda vez el valor máximo de feromona.

4. Conclusiones y trabajo futuro

La confianza es un aspecto fundamental en cualquier interacción entre nodos de una red P2P que tienen que trabajar en entornos dinámicos y, la mayoría de las veces, inciertos. Hoy por hoy, la ausencia de modelos precisos y rigurosos que gestionen de forma eficaz y eficiente la confianza en estos entornos supone un lastre que frena el avance y sobre todo las múltiples aplicaciones que se podrían dar a este tipo de redes.

En este trabajo se ha propuesto un modelo de confianza para redes P2P basado en el algoritmo de optimización bioinspirado de Colonia de Hormigas. Dicho modelo permite a un cliente de una red P2P que solicita un servicio buscar el camino "óptimo" que le conduzca a aquel nodo de la red que sea más confiable ofreciendo dicho servicio.

Se han planteado cuatro escenarios de prueba: uno en el que la bondad de todos los servidores es máxima, otro en el que dicha bondad es mínima, un tercero en el que la mitad de los servidores tienen bondad máxima y la otra mitad bondad mínima, y el último en el que todos los servidores tienen una bondad oscilante. Y como se ha visto en el apartado de análisis de resultados, éstos han sido bastante satisfactorios en todos los escenarios. Cabe decir que la bondad de un servidor determina cuánto se parecerá el servicio ofrecido por éste con el realmente proporcionado en la interacción con el cliente.

No obstante, a pesar de haber sido de los primeros en la aplicación de la técnica de SCH con obtención de buenos resultados, aún se pueden mejorar muchos aspectos del modelo. Es por ello que se proponen varios puntos como trabajo futuro.

En primer lugar se debería hacer una comparativa entre este modelo y sus equivalentes usando otros sistemas de optimización basados en hormigas como son el Sistema de Hormigas, el Sistema de Hormigas Elitista, el Sistema de Hormigas Mejor-Peor, el Sistema de Hormigas Max-Min o el propio Sistema de Colonia de Hormigas sin ninguna modificación.

También sería interesante comparar el modelo frente a otros modelos que no usaran optimización basada en hormigas, como por ejemplo los reseñados en la sección de estado del arte y algunos otros no mencionados en dicho apartado.

Como también se dijo en su momento, todos los parámetros utilizados finalmente en el modelo han sido optimizados a base de ensayo y error, probando distintos valores y quedándonos con aquellos con los que se obtenían mejores resultados. Estos parámetros, sin embargo, pensamos que aún pueden ser más óptimos y por ello se plantea tratar de mejorarlos como algo a conseguir en trabajos futuros.

Otro aspecto que consideramos podría estudiarse con más detalle con el fin de optimizarlo es la penalización aplicada a un servidor cuando proporciona un servicio insatisfactorio. Esta penalización es un elemento clave del modelo y por ello merece un análisis y estudio, si cabe, más riguroso.

De los cuatro escenarios planteados en la sección de análisis de resultados se ha observado que todos ellos tienen un comportamiento bastante bueno. Aún así, creemos que quizás sería posible mejorarlos un poco más, sobre todo algunos escenarios como el tercero. Se podría tratar de conseguir de alguna forma que la confianza de la mitad de los nodos que tienen bondad mínima no creciera o, al menos, creciera más lentamente.

Por supuesto, este modelo ha sido planteado desde su concepción inicial como una base que sirva de referencia a futuros modelos más completos que trabajen con redes P2P más complejas que las que nosotros hemos usado aquí. En definitiva, planteamos extender este modelo al escenario de la figura 1.(b), en el que hay múltiples nodos interconectados de cualquier manera entre ellos. Si en este escenario extendido el modelo sigue ofreciendo resultados tan buenos (y nosotros creemos que puede ser así), tendremos un modelo altamente competitivo y de gran calidad.

Por último, cabría destacar que se está trabajando en la redacción de un artículo para publicar en una revista de investigación que incluya la descripción de nuestro modelo, así como los resultados obtenidos con éste y el análisis de los mismos.

Referencias

1. G. Zacharia, P. Maes, "*Trust Management through reputation mechanisms*", Applied Artificial Intelligence, Vol. 14, 2000, pp 881-907.
2. J. Sabater, C. Sierra, "*REGRET: Reputation in gregarious societies*", Artificial Intelligence Research Institute, Spanish Scientific Research Council, 2001.
3. J. Sabater, C. Sierra, "*Social ReGreT, a reputation model based on social relations*", Artificial Intelligence Research Institute, Spanish Scientific Research Council, 2001.
4. J. Carbó, J.M. Molina, J. Dávila, "*Trust Management through Fuzzy Reputation*", Universidad Carlos III de Madrid y Universidad Politécnica de Madrid, International Journal of Cooperative Information Systems, Vol. 12, 2003, pp 135-155.
5. Yu Zhang, Huajun Chen y Zhaohui Wu, "*A Social Network-Based Trust Model for the Semantic Web*", University of Zhejiang, China, Autonomic and Trusted Computing, 3rd International Conference, ATC 2006, Springer, LNCS 4158, ISBN 3-540-38619-X.
6. Fajiang Yu, Huanguo Zhang, Fei Yan y Song Gao, "*An Improved Global Trust Value Computing Method in P2P System*", University of Wuhan, China, Autonomic and Trusted Computing, 3rd International Conference, ATC 2006, Springer, LNCS 4158, ISBN 3-540-38619-X.
7. Chenlin Huang, Huaping Hu y Zhiying Wang, "*A Dynamic Trust Model Based on Feedback Control Mechanism for P2P Applications*", National University of Defense Technology, Hunan, China, Autonomic and Trusted Computing, 3rd International Conference, ATC 2006, Springer, LNCS 4158, ISBN 3-540-38619-X.
8. Yun Tang, Lifeng Sun, Jianguang Luo, Shiqiang Yang y Yuzhuo Zhong, "*TPOD: A Trust-Based Incentive Mechanism for Peer-to-Peer Live Broadcasting*", University of Tsinghua, China, Autonomic and Trusted Computing, 3rd International Conference, ATC 2006, Springer, LNCS 4158, ISBN 3-540-38619-X.
9. Huirong Tian, Shihong Zou, Wendong Wang y Shiduan Cheng, "*A Group Based Reputation System for P2P Networks*", Beijing University of Posts and Telecommunications, China, Autonomic and Trusted Computing, 3rd International Conference, ATC 2006, Springer, LNCS 4158, ISBN 3-540-38619-X.
10. Ayman Tajeddine, Ayman Kayssi, Ali Chehab y Hassan Artail, "*PATROL-F - A Comprehensive Reputation-Based Trust Model with Fuzzy Subsystems*", Department of Electronical and Computer Enineering, American University of Beirut, Lebanon, Autonomic and Trusted Computing, 3rd International Conference, ATC 2006, Springer, LNCS 4158, ISBN 3-540-38619-X.
11. Tang Zhuo, Lu Zhengding y Li Kai, "*Time-Based Dynamic Trust Model Using Ant Colony Algorithm*", College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. Wuhan University Journal of Natural Sciences, Vol. 11, No. 6 2006, pp 1462-1466.
12. O. Cordón, F. Herrera y T. Stützle "*A Review on the Ant Colony Optimization Metaheuristic: Basis, Models and New Trends*", Mathware & Soft Computing, 9:2-3, 2002, pp 141-175.
13. M. Dorigo y L.M. Gambardella, "*Ant Colony System: A cooperative Learning Approach in the Traveling Salesman Problem*", IEEE Transaction on Evolutionary Computing, Vol. 1, No. 1, 1997, pp 53-66.