**Diseño e implementación de una regla de refuerzo para las hormigas de *forward* en AntNet**

**Design and implementation of a reinforcement rule for forward ants in AntNet**

Erick Abraham Roquel Pisquiy (roq17529@uvg.edu.gt)

Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica, Facultad de Ingeniería, Universidad Del Valle de Guatemala

**Resumen**

Este trabajo de investigación describe el diseño e implementación de una regla de refuerzo para las hormigas de *forward* en AntNet. Con pruebas preliminares se identificó la necesidad de implementar una penalización de feromona relacionada con las pérdidas vistas hacia determinado destino al utilizar como siguiente salto cierto vecino. Se añadieron estructuras y un algoritmo adicional a la implementación original para alcanzar los resultados. Se halló que al penalizar la feromona con base en el evento de pérdidas se puede mejorar la cantidad de veces que se puede hallar al destino deseado en un tiempo razonable, así como el tiempo en el que se halla la primera ruta completa y el *round trip time* final. Una métrica de rendimiento que no muestra mejora significativa es el tiempo de convergencia debido a oscilaciones introducidas por las pérdidas aleatorias del medio. En futuras implementaciones se puede identificar una variable que module el porcentaje de penalización de esta regla para mejorar los resultados.

**Palabras clave:** AntNet, problema de refuerzo, hormigas de *backward*, hormigas de *forward*, ruteo

**Abstract**

This work describes the design and implementation of a reinforcement rule for forward ants in AntNet. Preliminary tests revealed the necessity of applying a decrease in the pheromone for the neighbor that produced packet loss for certain destination. Structures and steps were added to the original AntNet version to accomplish this. The results revealed that using this rule it is possible to improve the percentage of times the algorithm finds a complete route for certain destination in reasonable time as well as the related round trip time. Also, it was possible to improve the final roud trip time obtained, but there was no meaningful enhancement in convergence time due to oscillations introduced by random packet loss. For future work it would be possible to identify and use a variable that modulates the decrease percentage to improve results.

**Keywords:** AntNet, reinforcement, backward ants, forward ants, routing

**Introducción**

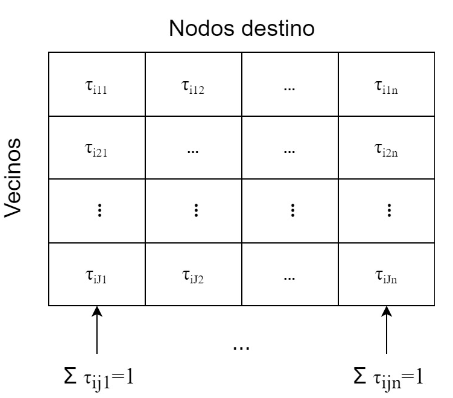
Los algoritmos de enrutamiento basados en inteligencia de enjambre aprovechan las cualidades de este tipo de inteligencia para resolver el problema de ruteo por medio de un algoritmo distribuido. De acuerdo con Kennedy (2006) la inteligencia de enjambre se basa en ciertos principios y reglas sencillas que incluyen: muchos individuos (un enjambre o *swarm*) y un solo objetivo, interacción entre los agentes y con su ambiente, reglas sencillas que definen las acciones que toman los agentes, toma decisiones descentralizada, aleatoriedad y el uso de un modelo de percepción – reacción. Este tipo de modelos permite que el agente evalúe cierta característica del fenómeno de interés (para el caso de este trabajo, el *round trip time* o RTT) y ejerza una función con base en esta observación para cumplir un objetivo global.

El problema de ruteo en una red de computadoras puede definirse según Kushwaha y Kumar (2014) como hallar un conjunto de saltos (*hops*) que debe tomar cierto paquete para llegar desde un nodo A hasta un nodo B. Según Juniper Networks (s.f.) se distinguen los términos ruta y camino (*path*) de manera tradicional como sigue: una ruta es una 2-tupla que consta de red de destino y siguiente salto, mientras que un camino es la sucesión de saltos tomada desde determinada fuente hacia cierto destino. Se dice que la tarea de ruteo es global ya que por medio de ellas se forman caminos entre diferentes fuentes y destinos, sin embargo a cada nodo “le interesa” que su tabla de rutas esté completa para los destinos de interés.

Las redes inalámbricas de sensores según Adamu (2013) son ejemplos de redes emergentes donde el problema de ruteo se distribuye a través de todos los nodos. En una red convencional, las redes contienen dos tipos de nodos que se conocen como *hosts*, quienes hacen uso de la red para cierta tarea a nivel de aplicación, mientras que los nodos de red llevan acabo la labor de ruteo. Para una red inalámbricas de sensores estas tareas suelen estar integradas, lo cual según Sohraby, Minoi y Znati (2007) demanda el diseño de nuevos protocolos de ruteo que puedan satisfacer los nuevos desafíos y requerimientos que presentan estas redes.

Los algoritmos de ruteo basados en inteligencia de enjambre, según Di Caro y Dorigo (2004) exhiben propiedades que pueden ser útiles para las redes inalámbricas que distribuyen la labor de ruteo en todos o varios de sus miembros. AntNet es un algoritmo que desarrolló Gianni Di Caro con base en Ant Colony que resuelve el problema de ruteo. Las ideas fundamentales de este algoritmo pueden describirse, según Markle y Middendorf (2006) como sigue: en intervalos regulares y conjuntamente con el tráfico, se envían hormigas de *forward* desde una fuente hacia un destino de interés, estas actúan simultáneamente y de manera independiente, comunicándose de manera indirecta por medio de la actualización de dos estructuras fundamentales que son la matriz de feromonas y un modelo de la red. El objetivo del modelo de la red es sentar las bases del modelo percepción – reacción implementado con base en el RTT, mientras que el rol de la matriz de feromonas es reflejar la “premiación” o penalización de cierta ruta con base en su desempeño a lo largo de la ejecución del algoritmo. Por otro lado, otro tipo de paquetes que se conocen como de *backward* se encargan de hacer efectiva la actualización de las estructuras en la red.

La matriz de feromonas tiene entradas que corresponden a la cantidad de feromona para cierto destino (corresponde a la columna) y cierto vecino que ejerce como siguiente salto para ese destino (corresponde a la fila). Esta estructura se ilustra en figura 1 donde se hace énfasis en el estándar de que la suma de las columnas sea igual a 1, lo cual quiere decir que las entradas se tratan como probabilidades, y la suma de todas las probabilidades de los elementos para ejercer como siguiente salto para un mismo destino debe ser igual al 100%. Las ecuaciones fundamentales que diseñó Gianni para la actualización de estas entradas se dan solamente en el evento de una hormiga de *backward*, “recompensando” la ruta atravesada como función del refuerzo r y con r = r(RTT). Estas se dan en (1) y (2).



**Figura 1.** Matriz T de feromonas con J el tamaño de la vecindad del nodo i.

El algoritmo original carece de eventos que actualicen la matriz T de feromona para las hormigas de forward. Según las simulaciones generadas por medio de RMASE (*Routing Modeling Application Simulation Environment*) para redes inalámbricas con una cantidad relativamente densa de vecinos (alrededor de 5 dentro del radio de alcance de la antena) puede tenerse problemas para hallar una ruta hacia cierto destino por la gran cantidad de decisiones que deben coincidir para que esto suceda. El objetivo de esta investigación es proponer una regla de actualización de la matriz T que se dispare en eventos relacionados con las hormigas de forward que sea capaz de tomar en cuenta las pérdidas de paquetes y de esta manera mejorar el descubrimiento de la ruta inicial y un mejor RTT final alcanzado.

**Materiales y métodos**

La plataforma base para las simulaciones es RMASE, la cual está escrita para MATLAB e integra un ambiente de simulación orientado a eventos discretos (*time-slotted*). La regla de asignación propuesta penaliza las rutas que generan pérdidas para cierto destino, con el objetivo de mejorar las métricas del algoritmo de ruteo. Se definen como métricas las siguientes:

* Retardo de arranque: es el tiempo que le toma al algoritmo hallar una primera ruta válida. Esto quiere decir que la hormiga de *backward* logró llegar exitosamente desde el destino de vuelta a la fuente. Esta métrica es importante ya que, a pesar de que la ruta inicial muy probablemente no es la óptima, habilita valores coherentes que pueden formar tablas de rutas para que el tráfico llegue de manera exitosa al destino.
* Retardo de convergencia: es el tiempo que le toma al algoritmo converger, lo cual se define como que existe una entrada en la matriz de feromonas T del nodo fuente para la columna correspondiente al destino de interés, que difiere de la unidad por menos de un umbral. Se define el umbral como 0.001.
* RTT de convergencia: es el RTT alcanzado para la ruta seleccionada en la condición de convergencia.

La regla de asignación propuesta se describe en (3) y (4)

Donde K = (1 - ) y L = . La ecuación (3) muestra la regla para el vecino que presenta pérdida de paquetes y al cual se quiere penalizar. Básicamente su feromona se reduce en un porcentaje r. La ecuación (4) presenta la regla de asignación para el resto de vecinos correspondientes a la columna del mismo destino y lleva a cabo una operación de normalización, garantizando que la suma de las columnas de la matriz de feromonas sea igual a 1.

**Gestionando las pérdidas**

*Cuantificación*

Para cuantificar las pérdidas existentes hacia un destino en particular se plantea la siguiente estructura que se almacena en memoria de cada agente del enjambre: una matriz S de 4 x N donde N es el número de destinos para la hormiga i. En la primera fila se almacena un contador que asciende cada vez que se envía una hormiga de *forward* hacia ese destino (correspondiente al número de columna), en la segunda un contador que asciende cada vez que se recibe una hormiga de *backward* desde ese destino, en la tercera un valor que almacena las pérdidas actuales, y en la cuarta el identificador del último vecino que ejerció como siguiente salto para el envió de la hormiga de *forward* a ese destino. Todos los valores de la matriz se inicializan en cero. El algoritmo que se presenta en el siguiente pseudocódigo:

*enviar\_forward(destino d, vecino v) or redirigir\_forward(destino d, vecino v):*

*perdidas\_historicas = S(3, d)*

*perdidas\_actuales = S(1, d) – S(2, d)*

*si perdidas\_actuales > perdidas\_historicas*

*penalizar(T, r)*

*S(1, d) = S(1, d) + 1*

*S(3, d) = perdidas\_actuales*

*S(4, d) = v*

Y en el caso que se reciba una hormiga de *backward* simplemente se aumenta la posición S(2, d) una unidad.

*Penalización*

La penalización se ilustra en el pseudocódigo de la sección anterior como *penalizar(T, r)*, lo cual denota una actualización de la matriz de feromonas T, por medio de las ecuaciones (3) y (4) y teniendo como parámetro el porcentaje de penalización r. La elección de r se hace de manera constante r = 0.05 de acuerdo con los resultados obtenidos en pruebas preliminares.

*Regla de refuerzo para hormigas de backward*

Para esta parte se utiliza una simplificación de la regla propuesta por Gianni, la cual limita el porcentaje de aumento del vecino “premiado” a 0.3. es el mejor valor de RTT obtenido durante las últimas w muestras, con w = 25.

**Escenario de pruebas**

A close-up of a calculator

Description automatically generated with low confidence

**Figura 2**. Topología de la red de pruebas

Para la realización de las simulaciones se utiliza un arreglo de 7 x 7 nodos, igualmente espaciados (1 unidad horizontalmente y 1 verticalmente), con radio de alcance por antena de 3 unidades. El criterio de paro de la prueba es una vez se alcance el criterio de convergencia. Si no se logra obtener una primera ruta al destino luego de 50 segundos se considera la iteración como fallida.

**Resultados y discusión**

**Comportamiento del algoritmo sin la regla propuesta**

A continuación, se presentan los resultados con el algoritmo estándar. Se hicieron diez corridas con resultado ilustrativos de la evolución de la feromona en el tiempo y el estado final de la ruta en figura 3. Los resultados cuantificados para este conjunto de pruebas están en tabla 1. El porcentaje de experimentos fallidos es alto (60%) lo cual muestra la dificultad con la que el algoritmo intenta hallar una primera ruta exitosa al destino. Esto tiene sentido ya que al inicio toda la probabilidad se distribuye de manera uniforme (no han actualizado las estructuras las hormigas de *backward)* por medio de la representación de la matriz de feromonas, así que varios eventos (saltos) deben coincidir para que el proceso sea exitoso. Como otra observación se ve que para la regla de refuerzo en *backward* utilizada, la primera ruta exitosa hallada suele ser la que va a converger al final, sin que se dé lugar a más exploración que perita obtener mejores rutas (la feromona para un vecino tiende a crecer sin que otras están cerca y el efecto de retroalimentación positiva del algoritmo hace que converga). Puede observarse en la última fila de figura 3 cómo el algoritmo convergió con muy poca “competencia” solo al inicio hacia esa ruta claramente no óptima.

Chart, line chart

Description automatically generatedDiagram

Description automatically generated with medium confidence

Chart, line chart

Description automatically generatedA picture containing diagram

Description automatically generated

Chart, line chart

Description automatically generated A picture containing graphical user interface

Description automatically generated

**Figura 3**. Evolución de la feromona en el tiempo y ruta convergente correspondiente para algoritmo original

**Tabla 1**. Métricas de rendimiento para algoritmo original.

|  |  |  |  |
| --- | --- | --- | --- |
| Porcentaje de pruebas exitosas | Retardo de arranque medio (ms) | Retardo de convergencia medio | RTT alcanzado medio |
| 40% | 24.39 | 45.98 | 0.2233 |

**Comportamiento del algoritmo con la regla propuesta**

Los resultados ilustrativos al implementar la regla de refuerzo para las hormigas de *forward* se muestran en figura 4. Se ve cómo las gráficas de evolución de feromona ya no son solo ascendentes como en el caso pasado, sino que oscilan levemente antes de llegar a converger. Esto tiene sentido ya que la ruta óptima también podría presentar pérdidas por las colisiones en el medio compartido. En algunos instantes otras rutas no óptimas tienen algunos destellos donde no decrecen o crecen levemente, pero, como se esperaba, las pérdidas hacia ese destino podrían ser mayores que con la ruta óptima, por lo que al elegirlas se penalizaría más, volviendo a tender a la ruta óptima. En las capturas de las rutas convergentes se ve como estas tienen más “sentido” o tienen una forma más similar a los que se esperaría sería la ruta óptima para esta topología. También es notable que se tienen mejores métricas de rendimiento, cabiendo destacar el parecido de la métrica de Retardo de convergencia medio comparado con el caso anterior. Esto quiere decir que, a pesar de que esta implementación podría mejorar el RTT alcanzado y el retardo de arranque, no necesariamente mejorará el retardo de convergencia. Esto tiene sentido ya que las penalizaciones a la ruta óptima en su asenso a la convergencia generan oscilaciones que se reflejan en mayor tiempo de convergencia.

**Chart, line chart

Description automatically generated**A picture containing polygon

Description automatically generated

**Chart

Description automatically generated** A picture containing diagram

Description automatically generated

Chart

Description automatically generatedDiagram

Description automatically generated with low confidence

**Figura 4**. Evolución de la feromona en el tiempo y ruta convergente correspondiente con la regla propuesta

**Tabla 2**. Métricas de rendimiento para algoritmo modificado.

|  |  |  |  |
| --- | --- | --- | --- |
| Porcentaje de pruebas exitosas | Retardo de arranque medio (ms) | Retardo de convergencia medio | RTT alcanzado medio |
| 50% | 11.19 | 44.59 | 0.1574 |

**Conclusiones**

La implementación de AntNet desarrollada en RMASE permitió simular el comportamiento del algoritmo al añadir una regla de refuerzo para las hormigas de *forward*. Esto fue útil para determinar que para el escenario de pruebas planteado y las iteraciones generadas, el algoritmo modificado tiene mejores métricas, siendo estas el retardo medio de convergencia, el RTT alcanzado y el retardo de arranque. Para el algoritmo original y el alterado se vio la tendencia de converger al primer camino hallado hacia el destino, sin embargo con el algoritmo modificado se tienen más oscilaciones de la feromona, lo cual significa que mejora le exploración de las soluciones. Como último punto, se vio que la mejora en la métrica de retardo de convergencia no era significativa, lo cual quiere decir que al usar esta regla propuesta, se podrá tener más rápido una ruta inicial (aunque no sea la óptima) y una ruta de convergencia de mejor calidad, aproximadamente en el mismo tiempo que el algoritmo original.

Como recomendaciones para trabajo futuro se podrían utilizar valores de r para la regla de *forward* que sea función de alguna variable que permita modularla, por ejemplo, las pérdidas históricas registradas hacia un destino por medio de determinado vecino. También pueden probarse otras topologías para comprobar si las tendencias vistas en este trabajo son replicables en todos los casos. Por último, se pueden explorar otras reglas de refuerzo de *backward* y su interacción con la regla de *forward* propuesta.

**Bibliografía**

Adamu, M. (2013). Energy-efficient routing algorithms based on swarm intelligence for wireless sensor networks,” Tesis doct., University of Nottingham.

Di Caro, G., Dorigo, M. (2004). Ant colony optimization and its application to adaptive routing in telecommunication networks Tesis doct., PhD thesis, Faculté des SciencesAppliquées, Université Libre de Bruxelles.

Juniper Networks. Getting started with networking, Juniper online education, Extraído de https://learningportal.juniper.net/juniper/user\_activity\_info.aspx?id=769

Kennedy, J. (2006). “Swarm intelligence,” en Handbook of nature-inspired and innovative computing, Springer: 187-219

Kushwaha, S. K., Kumar, A., Kumar, N. outing Protocols and Challenges Faced in Ad hoc Wireless Networks. (2014) Adv. Electron. Electric Eng, vol. 4: 207-212.

Merkle, D., Middendorf, M. (2006). Ant Colony Optimization, MIT Press ISBN: 0-262-04219-3.

Sohraby , K., Minoli , D., Znati, T. (2007). Wireless sensor networks: technology, protocols, and applications. John wiley & sons.