

**UNIVERSIDAD CENTROCCIDENTAL
“LISANDRO ALVARADO”**

**ALGORITMO DE COLONIA DE HORMIGAS PARA EL ENRUTAMIENTO
EN REDES DE SENSORES INALÁMBRICOS**

RODRIGO SAID HENRIQUEZ HERNÁNDEZ

Barquisimeto, 2011

UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”
DECANATO DE CIENCIAS Y TECNOLOGÍA
POSTGRADO EN CIENCIAS DE LA COMPUTACIÓN

ALGORITMO DE COLONIA DE HORMIGAS PARA EL ENRUTAMIENTO
EN REDES DE SENSORES INALÁMBRICOS

Trabajo de Grado presentado para optar al título de Magíster Scientiarum en Ciencias
de la Computación

Por: RODRIGO HENRIQUEZ HERNÁNDEZ

Barquisimeto, 2011

A la memoria de mi padre Rodrigo, que Dios lo tenga en su gloria.

A mi madre, Crucita.

Rodrigo S. Henríquez

AGRADECIMIENTO

Doy las gracias primeramente a Dios Todopoderoso, por iluminar mi camino y hacerme sentir la alegría de vivir confiadamente en él.

A mi padre, que aunque no estuvo presente desde pocos meses previo a la culminación de éste trabajo, siempre estuvo conmigo dándome ánimos y apoyo para lograr ésta meta.

A mi madre, por brindarme su amor y apoyo incondicional. Gracias mamá por tenerte a mi lado ayudándome a mantener la marcha.

A Sandra por su cariño y su apoyo en todo momento, dándome ánimos para mantener la constancia en la realización de éste trabajo.

A mis hermanas Diosymar y Rodesy, por su colaboración y comprensión durante el tiempo dedicado al estudio e investigación.

Al profesor José Gregorio Sánchez, por ser tutor de esta investigación y por su invaluable apoyo académico y moral.

A la profesora Niriaska, por sus consejos que sirvieron de gran estímulo para la culminación de éste trabajo.

Al profesor Luis Abraham, por su enseñanza y orientaciones metodológicas.

Al profesor Manuel Mujica, por su colaboración en la etapa final de la investigación.

A todas aquellas personas que de una manera u otra, contribuyeron a la culminación exitosa de este trabajo.

Mil gracias.

Que Dios les dé bendiciones, conforme a los deseos de su corazón...

Rodrigo S. Henríquez

ÍNDICE GENERAL

	PÁG.
DEDICATORIA	iii
AGRADECIMIENTO	iv
ÍNDICE DE FIGURAS.....	vii
ÍNDICE DE CUADROS.....	viii
RESUMEN.....	x
INTRODUCCIÓN.....	1
CAPÍTULO	
I EL PROBLEMA.....	4
Planteamiento del Problema.....	4
Objetivos de la investigación	11
<i>Objetivo General</i>	11
<i>Objetivos Específicos</i>	11
Justificación e Importancia	11
Alcance y Limitaciones	12
II MARCO TEÓRICO	14
Antecedentes de la Investigación	14
Bases Teóricas.....	18
Redes de Sensores Inalámbricos.....	18
Arquitectura de redes de sensores inalámbricos	18
Topología Estrella (Un punto a multipunto)	18
Topología Malla (Multipunto).....	19
Topología Híbrida.....	20
Algoritmos de Enrutamiento	21
Algoritmos Estáticos.....	23
Algoritmos Dinámicos	24
Consumo de energía de la red	26
Algoritmos de Optimización.	28
Algoritmos de Búsqueda Exhaustiva.....	28

	Algoritmos Heurísticos.....	29
	Algoritmos Metaheurísticos.....	32
	Algoritmos Evolutivos.....	34
	Metaheurística de Optimización de Colonia de Hormigas ...	35
	Comportamiento de las hormigas	38
	Algoritmo de Metaheurística OCH.....	41
III	MARCO METODOLÓGICO.....	44
	Tipo de Investigación	44
	Fases del Proyecto de Estudio.....	45
	Factibilidad Técnica.....	45
	Factibilidad Económica.....	46
	Procedimiento de la Investigación.....	46
IV	DISEÑO E INGENIERÍA DE LA PROPUESTA.....	47
	Simulación de la Red	47
	Distancia entre los nodos.....	51
	Representación de las Hormigas.....	51
	Rastros de Feromona.....	52
	Información total de feromona y Heurística.....	53
	Algoritmo de enrutamiento basado en Colonia de Hormigas	53
	Procedimiento Principal del Algoritmo de Colonia de Hormigas	55
	Construcción de los caminos	55
	Actualización local de los rastros de feromona	58
	Actualización de energía consumida.....	59
	Actualización global de los rastros de feromona	61
V	EXPERIMENTOS Y RESULTADOS.....	64
VI	CONCLUSIONES Y RECOMENDACIONES	73
	Conclusiones	73
	Recomendaciones	75
	GLOSARIO DE TÉRMINOS TÉCNICOS	76
	ANEXOS.....	78
	REFERENCIAS BIBLIOGRAFICAS.....	85

ÍNDICE DE FIGURAS

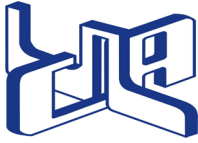
FIGURA		PÁG.
1	Topología de red Estrella	19
2	Topología de red de Malla	20
3	Topología Híbrida Malla y Estrella	21
4	Modelo del circuito de transmisión y recepción del Radio	26
5	Red de Sensores Inalámbricos de la red en topología de Malla	47
6	Diagrama de flujo del Algoritmo de Colonia de Hormigas aplicado al problema de Red de Sensores Inalámbricos.	54
7	Distribución espacial de los nodos en la red aleatoria de 100 sensores.	65
8	Energía consumida total en la ruta obtenida por la mejor hormiga	66
9	Media de los valores de energía de las rutas de las hormigas	67
10	Desviación estandar de los valores de energía de las rutas de las hormigas.	68
11	Distancia entre las hormigas con respecto a sus rutas	68
12	Distancia con respecto a la ruta entre las mejores hormigas.	69
13	Convergencia del algoritmo	70

ÍNDICE DE CUADROS

CUADRO		PÁG.
1	Algoritmo de la Metaheurística OCH.	42
2	Estructura de datos para la representación de los nodos de la red	50
3	Estructuras de datos para la representación del problema de enrutamiento en una RSI en la implementación del algoritmo OCH	50
4	Estructura de datos para la representación de las hormigas	52
5	Estructuras de datos para la representación de la matriz de feromona y el total de información heurística.	53
6	Procedimiento Principal del OCH para la RSI	55
7	Procedimiento para la construcción de las rutas	56
8	Procedimiento para la actualización local de feromona.	59
9	Procedimiento de actualización de consumo de energía de cada ruta.	60
10	Procedimiento para la actualización de feromona	62
11	Procedimiento para la actualización global de feromona.	62

ÍNDICE DE TABLAS

TABLA		PÁG.
1	Parámetros de la simulación del algoritmo	49
2	Valores de los parámetros de la simulación del OCH	64



ALGORITMO DE COLONIA DE HORMIGAS PARA EL ENRUTAMIENTO EN REDES DE SENSORES INALÁMBRICOS

Autor: Rodrigo Henríquez Hernández

Tutor: José Gregorio Sánchez

RESUMEN

Las Redes de Sensores Inalámbricos representan un área de investigación desafiante en la actualidad. Gracias a los avances que se han estado introduciendo en éstas redes, cada vez emergen más aplicaciones de monitoreo de ambiente y rastreo de objetos con sensores. En éstas redes, los sensores tienen capacidades limitadas de energía, procesamiento y almacenamiento de datos, las cuales se convierten en restricciones al momento en que se realiza la transmisión de la información adquirida a la estación central para su posterior procesamiento. Los algoritmos de enrutamiento para las Redes de Sensores Inalámbricos deben tomar en cuenta estas restricciones en el diseño con el fin de aumentar el rendimiento promedio de la red. En este sentido, el presente estudio desarrolla un algoritmo basado en la Metaheurística de Optimización basada en Colonia de Hormigas para el enrutamiento de datos en Redes de Sensores Inalámbricos configurados en una topología de Malla. Está basado en el comportamiento natural de las hormigas y se ha comprobado su eficacia en problemas de optimización donde la solución no se puede encontrar en tiempo polinomial. Mediante el uso de una regla probabilística que comprende una heurística y la actualización de feromonas, el algoritmo de colonia de hormigas encuentra rutas de bajo consumo de energía, aumentando así la vida útil de la red. La investigación se basa en Estudio de Proyectos, con soporte en el trabajo de campo y documental, lo que permite determinar diferentes configuraciones y capacidades de red con el fin de encontrar las “mejores” rutas o caminos. Los experimentos son realizados en un ambiente de simulación para evaluar el funcionamiento del algoritmo, dando resultados que demuestran su éxito en la búsqueda de rutas óptimas con bajo consumo de energía.

Palabras Claves: Colonia de Hormigas, Algoritmos de Optimización, Redes de Sensores Inalámbricos, Algoritmos Metaheurísticos.

INTRODUCCIÓN

Las redes de sensores inalámbricos son herramientas importantes en muchas aplicaciones tanto civiles como militares, donde son usados para recolectar, procesar y diseminar un amplio rango de información ambiental compleja. En los últimos años, se han convertido en un área de interés para la investigación. Dichas redes están compuestas por una serie de sensores que son usados para recolectar información del ambiente en varios tipos de aplicaciones, tales como supervivencia militar, monitoreo del hábitat, monitoreo industrial, agricultura de precisión, entre otras. Se encuentran desplegadas en diversos ambientes donde cada sensor es capaz de adquirir información y enviarla a otro nodo de la red hasta la estación base donde es almacenada para su posterior procesamiento.

Los sensores funcionan con batería y, por tanto operan con capacidades de energía extremadamente limitadas. Estas baterías, generalmente, son costosas y tienen una duración en el orden de meses o años, lo cual hace que el simple reemplazo de las mismas sea inadecuado en redes con miles de nodos integrados. Así, el consumo de energía se convierte en un factor importante que determina la vida útil del sensor y afecta el tiempo de operación de las redes inalámbricas.

La optimización de la energía, en el caso de las redes de sensores inalámbricos es compleja, debido a que involucra no solo al sensor, sino que además intenta maximizar el rendimiento de toda la red. El propósito de ésta investigación es implementar un algoritmo fundamentado en la Metaheurística de Optimización basada en Colonia de Hormigas para determinar rutas de transmisión de datos que permitan un consumo de energía mínimo en una red de sensores inalámbricos. Esta técnica de optimización ha sido exitosa en la resolución eficiente de problemas de optimización combinatoria (Dorigo y Stützle, 2004),

La meta deseada por la ejecución del algoritmo, es reducir las disipaciones generadas en el proceso de envío de los datos adquiridos por los sensores, con el objeto de asegurar la operatividad de la red por mayor tiempo. De esta manera se concreta la finalidad de esta investigación, la cual se orienta a implementar un algoritmo basado en la Metaheurística de Optimización de Colonia de Hormigas. El trabajo está dividido en seis (6) capítulos y está organizado de la siguiente manera:

El Capítulo I (El Problema) detalla la importancia del estudio de las Redes de Sensores Inalámbricos en problemas de la actualidad y a su vez justifica la realización de ésta investigación. Se enuncian los objetivos generales y específicos que se desarrollan y se exponen sus alcances y limitaciones.

El Capítulo II (Marco Teórico) contiene los antecedentes y las bases teóricas de las cuales toma lugar la investigación. Se presenta un estudio de las investigaciones realizadas en el enrutamiento de redes de sensores inalámbricos y los conceptos sobre algoritmos de resolución de problemas de optimización, entre los cuales se encuentra la metaheurística de optimización de colonia de hormigas, su definición, su comportamiento y los procedimientos del algoritmo.

En el Capítulo III (Marco Metodológico) se describe el tipo de investigación, el estudio de factibilidad y el procedimiento aplicado a largo de la investigación.

En el Capítulo IV (Diseño e ingeniería de la propuesta), se desarrolla la solución propuesta del algoritmo de Metaheurística de Colonia de Hormigas para una red de sensores inalámbricos.

En el Capítulo V (Experimentos y resultados), se presentan los experimentos realizados sobre simulaciones de redes de sensores inalámbricos, junto a comparaciones y análisis de los resultados obtenidos.

Finalmente, el Capítulo VI (Conclusiones y Recomendaciones), permite efectuar un cierre de todo el estudio, señalando las deducciones producto del proceso de la implementación del algoritmo, su desarrollo y experimentación en el lenguaje de programación utilizado, efectuando algunas disertaciones concretas que dan cabida a una serie de sugerencias que pueden ser tomadas por futuros investigadores en el área de las redes de sensores inalámbricos para el enrutamiento de datos en forma eficiente.

CAPÍTULO I

EL PROBLEMA

Planteamiento del Problema

La problemática en el uso de los sensores inalámbricos se da principalmente en las limitaciones de energía de sus baterías. El mayor consumo de energía se genera por las distancias en la transmisión y recepción de datos. En particular, en una topología de red malla, los sensores están dotados de una comunicación dada por retransmisiones entre sensores, lo cual disminuye las distancias locales de transmisión al momento de enviar la data a la estación base. Por tanto, es ventajoso que los nodos envíen la data a sensores que estén próximos para evitar gastar demasiada energía en grandes distancias y así mejorar el rendimiento promedio de la red. Sin embargo, la distancia no es la única variable determinante en la disipación de energía, ya que los nodos poseen diferentes cantidades de energía en la batería y pueden ser afectados por la retransmisión de los datos. En consecuencia, se hace necesario determinar una ruta óptima de transmisión de datos que consiga un balance entre la distancia de transmisión y la energía restante del sensor. Así, el problema de búsqueda de una ruta para la transmisión y recepción de datos con consumo mínimo de energía constituye un problema de optimización de un solo objetivo, el cual puede ser enfrentado mediante un algoritmo de enrutamiento.

Los sensores están siendo usados prácticamente en cualquier lugar y al ser usados en cantidad, ayudan a mantener fábricas en funcionamiento (Cao, 2009), optimizar la agricultura (The Memsic Solution, 2010), monitorar áreas (Martínez y Hart 2010) y hasta observar terremotos (Suzuki et. al. 2007). Estos dispositivos

permiten inspeccionar el ambiente y en conjunto proveen una útil herramienta, que al comunicarse entre sí, forman una red inteligente de adquisición de datos para ser procesados posteriormente en un análisis general del área monitoreada (Vázquez, 2007). Al ser integrados a estructuras, maquinarias, dispositivos, equipos, empresas y a diversas áreas, junto con la transmisión de la información adquirida, pueden proveer grandes beneficios a la sociedad: tolerancia a fallos, la conservación de recursos naturales, mayor productividad de manufactura, mejor respuesta a emergencias y sistemas de seguridad mejorados en el hogar, el comercio y la industria. (Lewis, F.L., 2004).

Sin embargo, existen barreras en el uso de sensores. Las grandes cantidades de cable y fibra óptica de las conexiones cableadas están sujetas a rupturas y fallas en los conectores. Las conexiones de largas cantidades de cable representan un costo en instalación y en mantenimiento a largo plazo, limitando el número de sensores que pueden ser desplegados, y por consiguiente, reduciendo la calidad total de la data reportada. Las redes de sensores inalámbricos pueden eliminar este costo, facilitando la instalación y eliminando conectores de cables.

En la actualidad, existen diferentes aplicaciones de las Redes de Sensores Inalámbricos (RSI). El monitoreo de áreas es una de las más comunes, en la cual se implementa en diversas regiones donde un fenómeno es objeto de seguimiento. Por ejemplo, es posible desplegar una gran cantidad de sensores en un campo de batalla para detectar intrusiones enemigas en lugar de utilizar minas terrestres (Dousse et. al., 2006).

Por otra parte, se han utilizado RSI en el control ambiental de vastas áreas de bosque o de océano para inspeccionar múltiples variables, como temperatura, humedad, fuego, actividad sísmica, entre otros. Por ejemplo, el monitoreo del estado

del permafrost en los Alpes suizos, entre los que se encuentran: el proyecto PermaSense (Hasler et. al., 2008) y el proyecto de Monitoreo de Glaciares (Martínez y Hart 2010).

En el sector industrial se pueden encontrar aplicaciones RSI dentro de áreas como el monitoreo del nivel de vertidos de tierra y conteo de bombas, seguimiento de llamas, control de nivel de torres de agua y tratamiento de aguas residuales, como es el caso del proyecto American Recovery and Reinvestment Act (Cao, 2009).

En la industria agrícola el uso de las RSI es cada vez más común. Conocida actualmente como agricultura de precisión, ésta tecnología es un método de gestión agrícola que permite al agricultor producir más eficientemente y obtener mayores beneficios económicos a través de un uso correcto de entrada de sus recursos. Entre las áreas que aborda la agricultura de precisión se encuentran la gestión del riego, protección contra las heladas, predicción de enfermedades, la ciencia y el monitoreo de cultivos de vivero. Ésta tecnología ofrece a los productores el acceso de forma inalámbrica a los datos relativos a su uso de agua, condiciones meteorológicas como temperatura y humedad. (The Memsic Solution, 2010)

Sin embargo, la problemática en el uso de los sensores inalámbricos es que poseen limitaciones de energía debido a que son mantenidos activos mediante el uso de baterías. Dicha energía es consumida por los procesos que ocurren dentro del sensor en la adquisición de los datos. Generalmente, el subsistema de radio requiere la mayor cantidad de energía. A mayor distancia de transmisión, mayor disipación de energía por el circuito de radio. Gracias a la capacidad de conectividad ad-hoc de los sensores, los datos pueden enviarse de un nodo a otro hasta llegar a una estación central donde posteriormente serán procesados. Por tanto, es ventajoso que los nodos envíen la data a sensores que estén próximos y con altos niveles de energía para

minimizar la transmisión y recepción de datos a grandes distancias y mejorar el rendimiento promedio de la red. Por tanto, la comunicación en una red de sensores inalámbricos debe considerar la energía disipada tanto por la transmisión y recepción como por el circuito electrónico del sensor.

En consecuencia, se hace necesario determinar una ruta óptima de transmisión de datos a través de la cual, los sensores envíen sus datos a vecinos cercanos con bajo nivel de energía o a lejanos con alto nivel de energía en la batería y así aumentar el tiempo de funcionamiento de la red. Por otra parte, una técnica de agregación y fusión de datos ayudaría a evitar encabezados de datos redundantes en la comunicación de red y en consecuencia, transmisiones de datos más cortas en tiempo. La combinación de estos métodos disminuye la cantidad de consumo de energía por transmisiones locales, que a largo plazo permite minimizar la disipación de energía global de la red. Esta forma de comunicación requiere que un agente de software visite cada nodo de la red e incrementalmente fusione data, minimizando a su vez, la energía consumida por el sensor durante dicho proceso.

Adicionalmente, en las redes de sensores inalámbricos existen distintos topologías de comunicación, las cuales son empleadas dependiendo de las necesidades específicas de la aplicación (Nakamura et. al. 2007). La topología estrella (descrito en las Bases Teóricas del Capítulo II), es una topología donde la estación base envía y recibe datos a todos los nodos, a los cuales no se les permite enviar mensajes entre sí. La ventaja de esta topología es la simplicidad y la de mantener un bajo consumo de energía en los sensores, no obstante, está limitada al rango de transmisión individual de los nodos. En la topología de malla, los nodos están dispuestos de tal forma que se interconectan unos con otros en un grafo totalmente conexo y a su vez se comunican con la estación central, permitiendo comunicaciones

de datos de nodo a nodo en una forma de múltiples saltos conocida en inglés como *multihop*. Esta topología de red tiene la ventaja de permitir ampliar el rango de la red; ya que no está necesariamente limitada al rango de los nodos individuales y puede ser ampliada simplemente añadiendo más sensores al sistema. La desventaja de este tipo de redes está en el consumo de energía de los sensores que implementan las retransmisiones en la comunicación y generalmente limitando la vida de la batería. También existen topologías híbridas que intentan aprovechar las ventajas de las topologías anteriores, sin embargo requieren que los sensores encargados de las retransmisiones sean provisionados de mejores capacidades de energía en la batería o si es posible tengan conexiones cableadas a la energía eléctrica.

En particular, para redes de sensores inalámbricos de bajo poder de batería, es suficiente el uso de la topología malla con una adecuada fusión de datos que permita transmisiones más cortas. En dicha fusión de datos, cada cierto tiempo la estación central realiza una petición de adquisición. Para ello, envía un paquete especial llamado agente (Al-Karaki, 2005), el cual contiene la ruta de desplazamiento, recorre todos los nodos, fusiona la data y regresa a la estación central para su posterior procesamiento.

Así, el problema de búsqueda de una ruta de desplazamiento para la transmisión y recepción de datos con consumo mínimo de energía en una red de sensores inalámbricos totalmente conectados, representa un problema de optimización de un solo objetivo, en el cual se intenta descubrir el mejor diseño o decisión.

Algunos algoritmos de búsqueda resuelven este problema de una manera determinista, buscando la mejor en todo el espacio de soluciones, lo cual representa su principal desventaja, pues su tiempo y espacio computacional puede llegar a ser

inaceptable. Generalmente, para este tipo de problema, es suficiente encontrar una solución parcial aproximada, donde se obtiene un mínimo local cercano al óptimo.

Existen alternativas exitosas como el caso de los algoritmos evolutivos que, utilizando el principio de evolución, producen aproximaciones muy cercanas a las soluciones óptimas mediante la supervivencia de individuos en nuevas descendencias. Sin embargo, también implican un costo computacionalmente alto.

La Metaheurística de Optimización de Colonia de Hormiga (OCH) ha sido usada eficazmente en la resolución de problemas de Optimización combinatoria (Dorigo y Stützle, 2004), y puede ser adaptada a problemas en los cuales no es posible encontrar soluciones en tiempo polinomial, en especial aquellos que comprenden búsqueda de rutas óptimas. Esto es posible gracias al uso de colonias de hormigas artificiales que simulan el comportamiento coordinado de las hormigas naturales en el proceso de búsqueda de alimento para llevarlo hacia el hormiguero. Un ejemplo es el caso del Problema del Viajero Comerciante TSP (por sus siglas en inglés Traveling Salesman Problem), el cual puede ser visto como un problema general cuya meta es conseguir un circuito Hamiltoniano de costo mínimo en un grafo de pesos.

Así, al intentar optimizar el tiempo de operación de una red de sensores inalámbricos totalmente conexa en términos de minimizar el consumo total de energía, surgen las siguientes interrogantes:

- ¿Es posible determinar rutas que minimicen el consumo de energía total de la red mediante la implementación de un algoritmo basado en la Metaheurística de Colonia de hormigas?

- ¿Cómo determinar las heurísticas a ser utilizadas por el algoritmo para resolver el problema de búsqueda de mejores rutas en términos de consumo de energía?
- ¿Cómo evaluar el funcionamiento del algoritmo en un ambiente de simulación de red de sensores inalámbricos para el enrutamiento de los datos?

Se espera que sea posible aplicar la Metaheurística de Optimización de Colonia de Hormiga al problema de enrutamiento en RSI configurados en malla, gracias a la capacidad que tienen dichos algoritmos en la búsqueda de rutas en un grafo de nodos. El propósito de ésta investigación está orientado a implementar un algoritmo basado en la Metaheurística OCH para determinar rutas de transmisión de datos que permitan un consumo de energía mínimo en una red de sensores inalámbricos.

Objetivos de la investigación

Objetivo General

Desarrollar un algoritmo basado en la Metaheurística de Optimización de Colonia de Hormigas para el enrutamiento de datos en Redes de Sensores Inalámbricos.

Objetivos Específicos

1. Implementar un algoritmo basado en la Metaheurística de Optimización de Colonia de Hormigas aplicado a una red de sensores inalámbricos configurados en una topología de Malla.
2. Determinar la información heurística y la cantidad de feromona requerida por el algoritmo de colonia de hormiga en la búsqueda de una ruta óptima en términos de energía para la comunicación en la red.
3. Evaluar el funcionamiento del algoritmo en un ambiente de simulación de red de sensores inalámbricos para el enrutamiento de los datos.

Justificación e Importancia

Recientemente se ha generado gran interés en el diseño, desarrollo e implantación de sistemas de sensores, lo que conduce a una creciente demanda en la interconexión de redes inalámbricas de sensores con otras tecnologías emergentes.

En la agricultura, la red de sensores inalámbricos crea una nueva perspectiva para el productor. Los tratamientos químicos sólo se utilizan cuando y donde sean

necesarios, la cantidad de agua gastada se mide de manera precisa y el uso de medidas preventivas se reduce al mínimo. Debido a que los costos de los insumos siguen aumentando, la necesidad de monitorear y administrar los recursos vitales como el agua es de suma importancia.

En el control ambiental, el monitoreo de múltiples variables, tales como temperatura, humedad, fuego y actividad sísmicas, son algunas de las que han sido posibles gracias a las Redes de Sensores Inalámbricos, lo cual ayuda a expertos a diagnosticar o prevenir problemas y minimiza a su vez el impacto ambiental de la presencia humana.

La demanda de RSI suele implicar algún tipo de vigilancia, seguimiento o control. Cuando los sensores detectan cambios en los niveles de calor, presión, sonido, luz, campo electro magnético, vibración, etc., el evento debe ser reportado a las estaciones base, la cual puede tomar una acción apropiada (por ejemplo, enviar un mensaje por Internet o a un satélite). Por ello es indispensable y necesario asegurar un óptimo funcionamiento de la red, en particular transmisión de datos, a fin de que ésta sea eficiente y durante la mayor cantidad de tiempo, sin fallas por agotamiento de la energía que pueda aportar las baterías. Esto solo se puede lograr a través del diseño de un algoritmo de enrutamiento que garantice la comunicación entre nodos con el máximo ahorro de energía.

Alcance y Limitaciones

En la presente investigación, el estudio se limita a la implementación del algoritmo de Metaheurística de OCH en una red de topología malla donde los sensores transmiten los datos directamente a la estación base o a cualquier nodo en la red, tal como se observa en los trabajos de Rajagopalan et al., (2005), a diferencia de

otros autores como Heinzelman W. et al (2000), Lindsey y Raghavendra. (2002) y (Camilo, et. al., 2006), en los que por el contrario, la información obtenida por los sensores es dirigida a la estación base por medio de los nodos intermedios llevando a cabo envíos de datos individuales.

La solución propuesta usa un agente hormiga que recorre todos los sensores fusionando la data recolectada y la envía a la estación base. Esta puede ser una limitante de aplicabilidad en casos de redes reales, en donde los sensores no dispongan de una visibilidad total de la red, es decir, poca cobertura. Por el contrario sería ventajoso en los casos donde todos los nodos tengan comunicación entre sí gracias a los caminos alternos con bajos niveles de energía que pueden ser obtenidos por el algoritmo, creando en consecuencia tolerancia a fallos.

Por otra parte, los sensores poseen restricciones en cuanto a la capacidad de energía en la batería tal como sucede en la vida real. Sin embargo, se asumen homogéneos, lo cual puede diferir a ciertos casos de la actualidad donde los sensores poseen distintas características. El modelo de energía gastada por los sensores considera la cantidad de energía disipada por el circuito electrónico y por el amplificador de transmisión. Por un lado la cantidad de energía que se gasta en la transmisión varía con respecto a la distancia y la cantidad de energía electrónica, y la cantidad de energía gastada en la recepción considera únicamente el circuito electrónico.

Las simulaciones se ejecutaron vía software utilizando distintos modelos de redes de sensores inalámbricos para analizar el desempeño del algoritmo de enrutamiento. Dicho algoritmo está basado en la Metaheurística de OCH aplicado al problema de redes de sensores inalámbricos. Es indispensable mencionar también, que durante la ejecución del algoritmo no se modifica la topología de la red.

CAPÍTULO II

MARCO TEÓRICO

Antecedentes de la Investigación

Las Redes de Sensores Inalámbricos están diseñadas para la recolección de información del ambiente mediante una serie de nodos autónomos que envían su información a una estación base. Estos sensores presentan recursos limitados de almacenamiento de energía y en sus operaciones de envío y recepción de datos de forma inalámbrica se reduce el tiempo de operación de la red.

En los algoritmos de enrutamiento tradicionales no se tomaba en cuenta las restricciones de energía de los nodos tal como en trabajos realizados por Perkins C. et al. (1994, 1999). Este hecho disminuye en gran medida el tiempo de vida de la red. Por tal motivo, se han diseñado nuevos protocolos de enrutamiento que intentan minimizar el consumo total de energía para así aumentar el tiempo de vida de la red.

Para una arquitectura de redes RSI basada en clústeres, comúnmente se han utilizado los siguientes protocolos de enrutamiento: Jerarquía de Clústeres Adaptables de Baja Energía LEACH (por sus siglas en inglés Low-Energy Adaptive Clustering Hierarchy) desarrollado por Heinzelman W. et al (2000) y el protocolo Reunificación de Ahorro de Energía en Sistemas de Información del Sensor PEGASIS (por sus siglas en inglés Power-Efficient Gathering in Sensor Information Systems) desarrollado por Lindsey y Raghavendra (2002). Estos algoritmos intentan minimizar el consumo de energía de la red mediante la elección de clústeres. El primero elige de manera estocástica los clústeres de la red, mientras que el segundo los selecciona de

tal manera que los sensores queden organizados en forma de cadena hasta la estación base.

En el protocolo LEACH presentado por Heinzelman W. et. al. (2000), se forman varios clústeres en una manera organizada donde se designa aleatoriamente un nodo como cabecera del clúster. Estos nodos se diferencian de los demás sensores por tener mayores capacidades de procesamiento, almacenamiento y poder en la batería. Cada nodo envía la data capturada a los nodos cabecera y éstos a su vez envían la data entre ellos hasta llegar a la estación base. Debido a que las transferencias a la estación base disipan mucha energía, los nodos toman un turno para la transmisión mediante la rotación de nodos cabecera. Esta rotación, lleva a un consumo balanceado de energía de todos los nodos y en consecuencia, a un tiempo de vida más largo de la red. La operación de LEACH se divide en dos (2) rondas, una de ellas es de configuración y la otra es la fase de estado sólido. Durante la fase de configuración se determinan los nodos cabecera del cluster, donde un nodo se elige a sí mismo como cabecera al determinar un valor aleatorio entre 0 y 1. Si el número sobrepasa un umbral, el nodo se designara a sí mismo como el nodo cabecera durante la ronda actual y le envía un mensaje a los nodos vecinos para que se unan al cluster. Durante la fase de estado sólido, se transfiere los datos a la estación base. La selección estocástica de los nodos cabeceras en LEACH no contribuye al consumo mínimo de energía durante la transferencia para un conjunto dado de sensores.

El protocolo PEGASIS presentado por Lindsey y Raghavendra (2002), requiere menos energía que el protocolo LEACH, debido a que forma una cadena de sensores, donde cada sensor envía la data a un vecino cercano. La data se fusiona de nodo a nodo hasta que un nodo envíe a la estación base. Cada vez que los sensores toman información del ambiente, toda la data de cada uno de los nodos deben ser enviados a

la estación Base, donde se realizará el procesamiento o almacenamiento de la data obtenida. Debido a que los nodos se encuentran a distintas distancias lejos de la estación base, la cantidad de energía necesaria para enviar la data por los sensores más lejanos es muy alta, por lo que llevarían al nodo a una rápida pérdida de poder en la batería. Gracias a la capacidad de comunicación ad-hoc de las redes inalámbricas, cada nodo puede enviar un paquete a los nodos adyacentes, realizando una fusión de la data recolectada entre cada uno de los nodos hasta llegar a la estación base. La fusión de datos ayuda a reducir la cantidad de energía gastada total, al transmitir la data entre sensores y la estación base, combinando uno o más paquetes de las mediciones de distintos sensores para producir un solo paquete.

Tanto LEACH como PEGASIS utilizan fórmulas de energía que les permiten obtener el menor número de retransmisiones o saltos para la determinación de la ruta, a nivel de cabeceras de clúster para el primer protocolo, y cadenas de nodos en el segundo. Por su parte, EEABR es un protocolo de comunicaciones para Redes de Sensores Inalámbricos, llamado así por las siglas de la traducción al inglés de Algoritmo de Enrutamiento Eficiente en Energía Basado en Hormiga (Camilo, et. al., 2006), el cual está basado en la metaheurística de Optimización de Colonia de Hormiga. EEABR usa una colonia de hormigas artificiales para viajar por la red de sensores para buscar rutas entre los nodos sensores y un nodo destino, que es a su vez corto en longitud y eficiente en energía, contribuyendo en tal manera a maximizar el tiempo de vida de la RSI. El algoritmo EEABR muestra buenos resultados en distintos escenarios, sin embargo, no considera la correlación de datos y el consumo de energía en la comunicación es muy alto cuando existe una gran cantidad de fuentes de envío de datos en la red.

E&D ANTS (Wen, et. al. 2007), es un algoritmo adaptativo cuyo modelo es *Energía*Retardo* y está basado en Optimización de Colonia de Hormiga. Este algoritmo está diseñado para mejorar el rendimiento en una Red de Sensores Inalámbricos distribuida aleatoriamente, mediante la búsqueda de una ruta óptima que maximice el tiempo de vida de la red y que además provean servicios de transmisión de datos en tiempo real. La idea de éste algoritmo es minimizar el retardo al transferir un número fijo de datos con restricciones de energía. Para el entrenamiento del modelo planteado, éste algoritmo usa Aprendizaje por Refuerzo, lo cual demuestra robustez. Sin embargo cuando la carga de tráfico de red es alta, se requieren retransmisiones, lo cual consume gran cantidad de energía debido a la alta pérdida de paquetes.

Aunque se ha probado la efectividad de algoritmos que intentan optimizar el tiempo de operación de la red, ninguno de ellos puede conseguir el valor mínimo de consumo de energía que lleve al máximo el tiempo de operación de la red, en particular, si los nodos están dispuestos en una topología de malla.

El presente trabajo de investigación se diferencia notablemente de los estudios antecedentes previos, porque desarrolla un algoritmo de enrutamiento que aprovecha las bondades del comportamiento de hormigas en una arquitectura RSI totalmente conexas no jerárquica, es decir, sin presencia de nodos cabeceras de clúster. Considera la minimización del consumo de energía basada en información heurística que permite examinar diferentes rutas, sin limitarse a un número de saltos o retransmisiones. Su meta consiste en hallar la mejor ruta entre todas las posibles en un tiempo razonablemente aceptable y con bajos niveles de procesamiento computacional.

Bases Teóricas

Redes de Sensores Inalámbricos

Una red de sensores inalámbricos (RSI) en general consiste de una estación base que puede comunicarse con un número de sensores inalámbricos vía enlace de radio. La data es recolectada en el sensor inalámbrico, comprimida, y transmitida a la estación base directamente o, si se requiere, mediante retransmisiones usando otros sensores inalámbricos. La data transmitida es entonces presentada al sistema por la conexión a la estación base.

Un factor clave en cualquier sensor inalámbrico es minimizar la energía consumida por el sistema. Generalmente, el subsistema de radio requiere la mayor cantidad de energía. Por consiguiente, es ventajoso enviar data sobre el radio de la red de una forma eficiente que minimice la energía disipada en las transmisiones.

Arquitectura de redes de sensores inalámbricos

Existen distintas topologías para las redes de comunicación inalámbricas. Brevemente se dará una discusión acerca de las topologías de redes que aplican a las redes de sensores inalámbricos.

Topología Estrella (Un punto a multipunto)

Una red estrella (Figura 1) es una topología donde la estación base puede enviar o recibir un mensaje a un número determinado de nodos. Los nodos solamente pueden enviar o recibir un mensaje de la estación base y no se les permite enviar mensajes entre sí. La ventaja de éste tipo de redes es su simplicidad y la capacidad de mantener el consumo de los nodos remotos al mínimo. Esto permite comunicaciones de baja latencia entre el nodo remoto y la estación base.

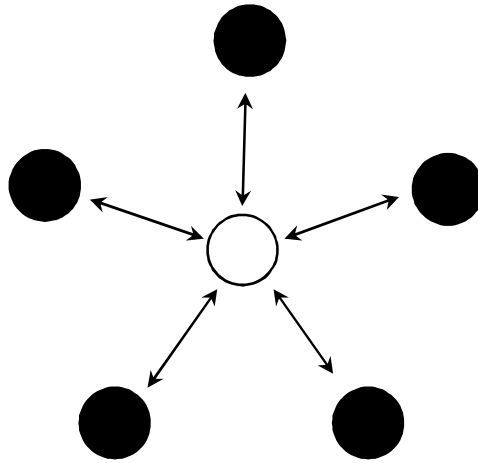


Figura 1: Topología de red Estrella

La desventaja es que la estación base debe estar dentro del radio de transmisión de todos los nodos individuales y no es robusto como otras redes debido a la dependencia a que un solo nodo administre la red.

Topología Malla (Multipunto)

La topología malla permite que un nodo en la red transmita a cualquier otro nodo que esté dentro del radio de transmisión (Figura 2). Esto permite lo que conocemos como comunicaciones multipunto; es decir si un nodo quiere enviar un mensaje a otro nodo que esté fuera de su rango de radio de comunicación, puede usar un nodo intermedio para retransmitir el mensaje al nodo deseado.

La ventaja de esta topología de red que agrega redundancia y escalabilidad. Si un nodo y en el camino de comunicación falla, un nodo x aún puede comunicarse a cualquier otro nodo z en su rango, y éste a su vez puede retransmitir el mensaje al nodo deseado. Adicionalmente, el rango de la red no necesariamente está limitado al rango entre los nodos individuales sino que puede ser extendido simplemente agregando más nodos al sistema.

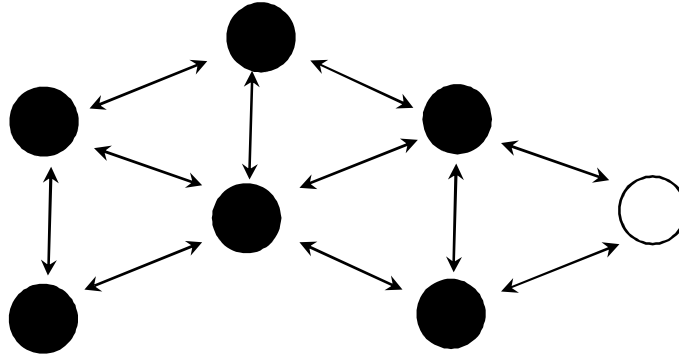


Figura 2: Topología de red de Malla.

La desventaja de este tipo de red está en que el consumo de energía de los nodos que implementan la retransmisión son generalmente mayor que los nodos que no tienen esta capacidad, limitando así la vida de la red. Adicionalmente, mientras el número de saltos al destino se incrementa, el tiempo para enviar el mensaje también se incrementa, especialmente si es un requerimiento que hayan operaciones de bajo poder de los nodos.

Topología Híbrida

En esta topología de red, ciertos sensores no están capacitados con la habilidad de retransmitir mensajes (Figura 3). Esto permite que sea mantenido un mínimo consumo de energía. Sin embargo, algunos nodos de la red están habilitados con capacidad de retransmisión, permitiéndoles a los demás nodos reenviar mensajes. Generalmente, los nodos con la capacidad de retransmisión están dotados de más alta energía y si es posible, están conectados a la línea eléctrica.

La topología Híbrida resulta útil en los casos donde el diseñador cuenta con los recursos necesarios para la instalación de diferentes tipos de sensores y de alto rendimiento en la retransmisión de mensajes.

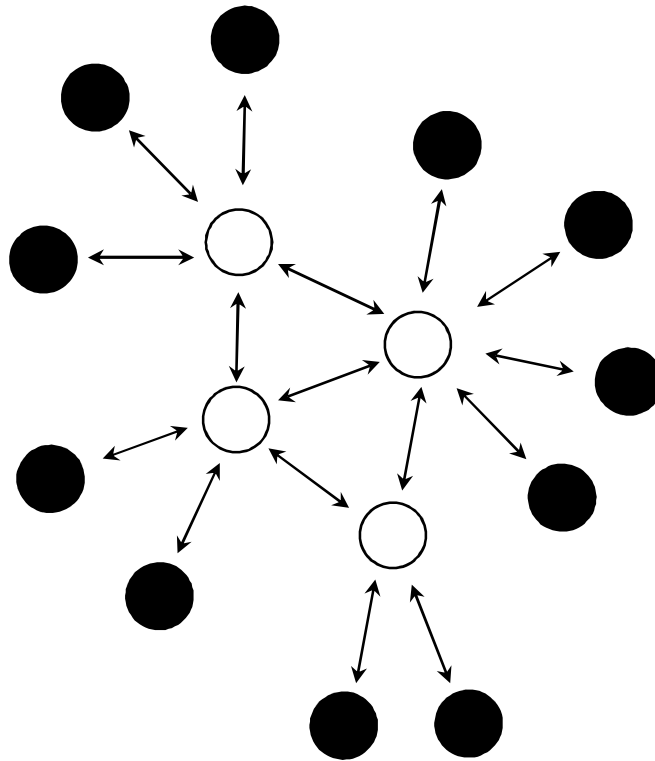


Figura 3: Topología Híbrida Malla y Estrella.

Las desventajas de ésta topología son similares a la topología malla pero en una forma reducida ya que al existir retransmisiones, la energía de los nodos de alta capacidad tiende a ser reducida a medida que se incrementa el número de mensajes que envían los sensores de baja capacidad.

Por consiguiente, en cualquier topología de redes de sensores inalámbricos es necesario el uso de un algoritmo que minimice el consumo de energía generado por el envío y recepción de los datos adquiridos.

Algoritmos de Enrutamiento

En una red de equipos, el algoritmo de enrutamiento es el que se encarga de llevar los datos desde un origen hasta un destino. Es posible que para que esto ocurra,

se requieran de muchos saltos por enrutadores intermedios. Un enrutador es un equipo que se encarga de retransmitir los datos hasta el equipo terminal. Al llegar los datos al enrutador por una entrada, éste debe escoger la salida a la cual debe enviarla, lo cual puede ser un nodo de su misma red al que puede acceder directamente o al de una red donde otro enrutador pueda retransmitir los datos al nodo terminal. Los nodos terminales solo envían y reciben los datos y no determinan que ruta debe escoger los datos en la red.

En una red de sensores inalámbricos, los sensores pueden cumplir roles tanto de equipo terminal como de enrutador. En una topología de red malla, los sensores inalámbricos deben retransmitir los datos mediante el uso de un algoritmo de enrutamiento que determine la ruta a seguir.

Los algoritmos de enrutamiento se pueden dividir en dos grupos: adaptables y no adaptables. Los algoritmos no adaptables no basan sus decisiones de enrutamiento en mediciones o estimaciones del tráfico ni en la topología. La decisión de qué ruta tomar de i a j se calcula por adelantado, fuera de línea y se cargan en los enrutadores al iniciar la red. Éste procedimiento se llama enrutamiento estático. Por lo contrario, los algoritmos adaptables cambian sus decisiones de enrutamiento para reflejar los cambios de topología y de tráfico. Difieren de los algoritmos estáticos en el lugar de obtención de su información (ej. localmente, en los enrutadores adyacentes o de todos), el momento del cambio de sus rutas (ej. cada Δt seg., o cuando cambia la carga) y la métrica usada para la optimalidad (ej. distancia, nº de escalas, tiempo estimado del tránsito). Éste procedimiento se llama enrutamiento dinámico. Este tipo de algoritmos no pueden ser demasiado complejos ya que son implementados en los enrutadores y deben ejecutarse en tiempo real con recursos de procesamiento y la cantidad de memoria con que dispone el enrutador. A

continuación se presentan algunos algoritmos de enrutamiento, tanto estáticos como dinámicos comúnmente usados.

Algoritmos Estáticos

Una de las técnicas estáticas muy conocida es el *enrutamiento por trayectoria más corta*, en el cual se arma un grafo de la subred donde cada nodo representa un enrutador y cada arco del grafo una línea de comunicación (enlace). Para seleccionar la ruta entre un par dado de enrutadores, el algoritmo simplemente encuentra en el grafo la trayectoria más corta entre ellos. El concepto de trayectoria más corta se debe a que la forma de medir la longitud de la ruta es usando alguna métrica (peso relativo que se da a los factores que intervienen en el cálculo de la distancia en una red) tal como el número de saltos, la distancia física, el retraso de transmisión, ancho de banda, el tráfico promedio, el costo de comunicación, entre otros.

El algoritmo más común de enrutamiento por trayectoria más corta es el Dijkstra (1959), el cual funciona de la siguiente manera:

- Cada nodo se etiqueta con su distancia al nodo de origen a través de la mejor trayectoria conocida.
- Inicialmente no se conocen trayectorias, por lo que todos los nodos tienen la etiqueta infinito.
- A medida que avanza el algoritmo y se encuentran trayectorias, pueden cambiar las etiquetas, reflejando mejores trayectorias.
- Una etiqueta puede ser tentativa o permanente.
- Inicialmente todas las etiquetas son tentativas.
- Al descubrirse que una etiqueta representa la trayectoria más corta posible del origen a ese nodo, se vuelve permanente y no cambia más.

El *enrutamiento basado en flujo* es otra técnica estática que usa tanto la topología como la carga para el enrutamiento (Bertsekas y Gallager 1992). Para una línea dada, si se conocen la capacidad y el flujo promedio, es posible calcular el retardo promedio de los paquetes en esa línea a partir de la teoría de colas. De los retardos promedio de todas las líneas, es directo el cálculo de un promedio ponderado por el flujo para obtener el retardo de paquete medio de la subred completa. El problema de enrutamiento se reduce entonces a encontrar el algoritmo de enrutamiento que produzca el retardo promedio mínimo para la subred.

Algoritmos Dinámicos

Uno de las técnicas dinámicas más comunes es el *enrutamiento vector de distancia*. Este algoritmo opera haciendo que cada enrutador mantenga una tabla (por ejemplo, un vector) que da la mejor distancia conocida a cada destino y la línea (interfaz) a usar para llegar ahí. Estas tablas se actualizan intercambiando información con vecinos. Este algoritmo recibe otros nombres como: algoritmo de enrutamiento Bellman-Ford distribuido y el algoritmo Ford-Fulkerson (Bellman, 1957; Ford y Fulkerson, 1962). Cada enrutador mantiene una tabla de enrutamiento indexada por, y conteniendo un registro de, cada enrutador de la subred. Esta entrada comprende dos partes:

- La línea preferida de salida hacia ese destino
- Una estimación del tiempo o distancia a ese destino.

La métrica usada podría ser la cantidad de saltos, el retardo de tiempo en milisegundos, el número total de paquetes encolados por la trayectoria, o una combinación de las mismas.

El enrutamiento por vector de distancia se usó en ARPANET hasta 1979, cuando fue reemplazado por el *enrutamiento por estado de enlace*. El primer concepto de enrutamiento por estado de enlace fue publicado por John M. McQuillan, (1979), como un mecanismo que pueda calcular rutas de una forma más rápida cuando cambian las condiciones de la red, y así llevar a cabo un enrutamiento más estable. Este algoritmo es usado en el protocolo OSPF (por sus siglas como *Open Shortest Path First*, descrito en Coltun, R, 2008), el cual es empleado con frecuencia en internet. El concepto del algoritmo de enrutamiento por estado de Enlace es sencillo y puede describirse en cinco partes. Cada enrutador debe:

- Descubrir a sus vecinos y conocer sus direcciones de red.
- Medición del costo de la línea
- Construcción de los paquetes de estado de enlace.
- Distribución de los paquetes de estado de enlace.
- Cálculo de nuevas rutas.

A medida que crece el tamaño de las redes, crecen proporcionalmente las tablas de enrutamiento, lo que consume memoria del enrutador, tiempo de procesamiento al examinarlas y ancho de banda al enviar informes de estado. En cierto momento, la red puede crecer hasta el punto en que ya no es factible que cada enrutador tenga una entrada para cada uno de los demás enrutadores.

Otro algoritmo de enrutamiento dinámico es el *enrutamiento por multidifusión*, tal como el presentado por Ballardie, A. (1997). Para la multidifusión se requiere de administración de grupo. Se necesita alguna manera de crear y destruir grupos, y un mecanismo para que los procesos se unan a los grupos y salgan de ellos. Para realizar el enrutamiento, cada enrutador calcula un árbol de expansión que cubre

a todos los demás enrutadores de la subred. Cuando un proceso envía un paquete de multidifusión a un grupo, el primer enrutador examina su árbol de expansión y lo recorta, eliminando todas las líneas que conduzcan a hosts que no sean miembros del grupo. Los paquetes de multidifusión se reenvían solo a través del árbol de expansión apropiado.

Consumo de energía de la red

En el presente trabajo, se asume un modelo de radio simple donde se disipa una energía para ejecutar el circuito amplificador de transmisión de $\epsilon_{amp} = 100 \text{ pJ/bit/m}^2$ y para ejecutar el circuito transmisor o receptor de $E_{elec} = 50 \text{ nJ/bit}$ (Heinzelman, et. al., 2000).

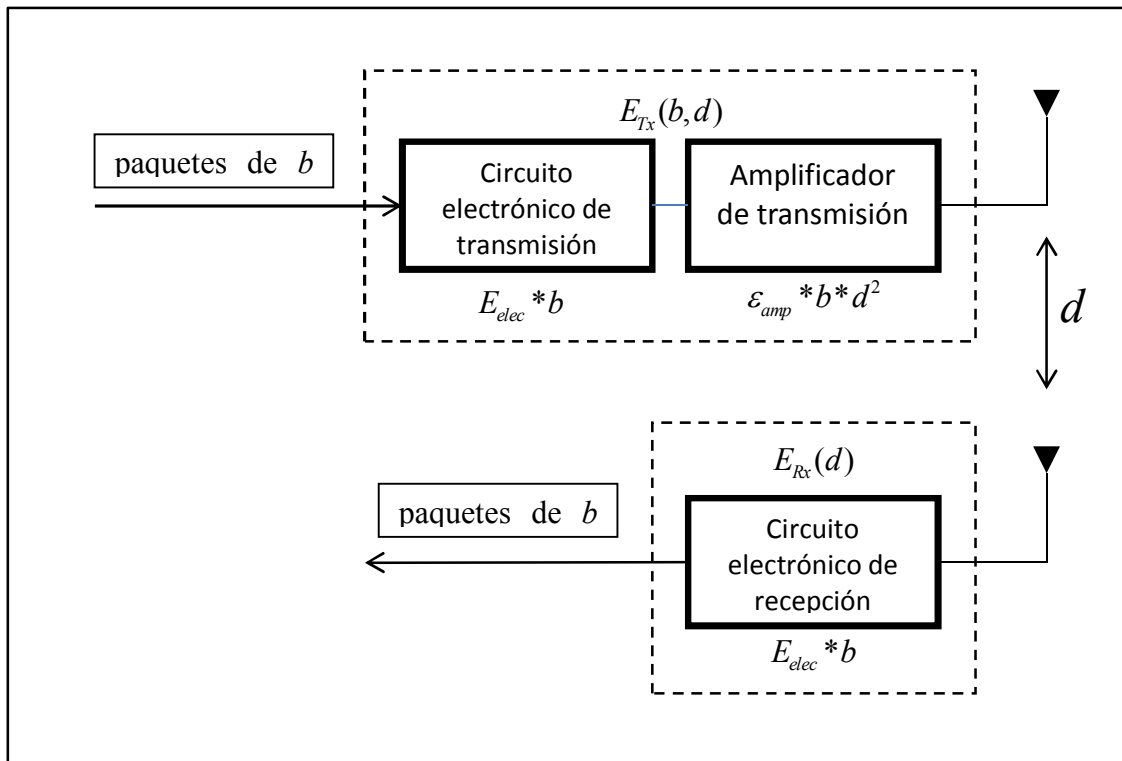


Figura 4. Modelo del circuito de transmisión y recepción del Radio

Se asume adicionalmente una pérdida de energía debida a la transmisión, la cual es proporcional al cuadrado de distancia euclidiana que existe entre los nodos (Figura 4).

Las ecuaciones de energía usadas en la transmisión (E_{tx}) y recepción (E_{rx}) de un mensaje, tal como en la literatura de R. Steele (1992) y Rappaport (1996), son las siguientes:

En la transmisión:

$$E_{tx} = E_{elec} * b + \epsilon_{amp} * b * d^2 \quad (2.1)$$

En la recepción:

$$E_{rx} = E_{elec} * b \quad (2.2)$$

donde:

E_{elec} : es la energía consumida en el circuito de recepción o transmisión.

ϵ_{amp} : es la energía consumida por el circuito amplificador de transmisión.

b : es la data que se recibe o envía durante la transmisión.

d : es la distancia entre los sensores.

Los b bits transmitidos están compuestos por la información del agente y la data de cada sensor, y va creciendo a medida en que se fusiona con la data de medición del siguiente sensor. La distancia d , es la distancia euclidiana entre el sensor emisor y el sensor receptor.

El proceso de recibir y transmitir un mensaje es proceso que tiene un alto costo como se puede observar en las formulas (2.1) y (2.2). El algoritmo debe intentar

minimizar no solo las distancias de las transmisiones escogiendo nodos cercanos sino también el número de operaciones de transmisión y recepción de cada mensaje.

Algoritmos de Optimización.

Los problemas de optimización hoy día tienden a ser cada vez más complejos, ya que presentan análisis y procesamiento de una gran cantidad de datos. En la búsqueda de la solución a estos problemas se presenta el desafío de encontrar una respuesta acertada en un corto plazo. Existen varios algoritmos de búsqueda que enfrentan estos problemas de una manera determinista, buscando todas las soluciones posibles y tomando luego la mejor solución. Aún si se pudiera desarrollar un algoritmo exacto, su tiempo y espacio computacional puede llegar a ser inaceptable. Es por ello que con frecuencia es suficiente conseguir una solución parcial aproximada. Los algoritmos heurísticos son capaces de encontrar aproximaciones muy cercanas a las soluciones óptimas.

Algoritmos de Búsqueda Exhaustiva

Los Algoritmos de Búsqueda Exhaustiva son aquellos que intentan buscar todas las soluciones posibles para un conjunto determinado y consecuentemente selecciona la mejor. Entre los algoritmos de búsqueda exhaustivas se tienen los siguientes:

- a) Búsqueda Local: Es un algoritmo de búsqueda exhaustiva que se enfoca en un área determinada del espacio de búsqueda. Uno de los algoritmos de búsqueda local más conocidos es el de escalamiento de colina (conocido en inglés como *hill-climbing*). En tales algoritmos, se reemplaza consistentemente la solución actual con la mejor de sus vecinos si es mejor que la actual.

- b) Algoritmos Dividir y Conquistar: en éstos se divide un problema en problemas más pequeños que sean más fáciles de resolver. Las soluciones pequeñas se combinan para encontrar una solución a la original. Ésta técnica es efectiva pero su uso es limitado porque no hay un gran número de problemas que puedan ser fácilmente particionados y combinados.
- c) La técnica de ramificación y acotación: ésta enumera el espacio de búsqueda, pero intenta colocarle reglas a partes del espacio de búsqueda que no pueden contener la mejor solución.
- d) Programación Dinámica: es una búsqueda exhaustiva que formula la solución como un proceso recursivo, evitando los cálculos repetidos al almacenar las soluciones a cada subproblema.

Las técnicas ramificación y acotación y programación dinámica son bastante efectivas pero con frecuencia el tiempo de cómputo es muy alto e inaceptable para tareas NP-completas. El algoritmo de escalamiento de colina es efectivo, pero tiene una debilidad significativa llamada convergencia prematura debido a que siempre consigue la solución óptima local de baja calidad. Es en éstos casos donde las técnicas heurísticas son más efectivas.

Algoritmos Heurísticos.

La heurística es una técnica usada para solucionar un problema, o más precisamente, para buscar una solución entre todos los posibles problemas, pero no garantizan que sea la mejor, por lo que puede considerarse como algoritmos aproximados. Tales algoritmos buscan de una manera fácil y rápida una solución cercana a la mejor. En algunos casos pueden ser precisos, es decir que pueden

conseguir la mejor solución, pero siguen siendo llamados heurísticos hasta que la mejor solución sea probada como la mejor de todas las soluciones.

Usualmente los algoritmos heurísticos son usados para resolver problemas que no pueden ser resueltos fácilmente. De acuerdo a la “dureza”, los problemas pueden distinguirse en clases de complejidad de tiempo. Los de clase P consisten en todos aquellos problemas que pueden ser resueltos en una máquina de Turing determinista en tiempo polinomial. Las máquinas de Turing son una abstracción usada para formalizar la noción de algoritmo y complejidad computacional (Garey y Johnson, 1979). Los de clase NP consisten en todos aquellos problemas cuya solución puede ser encontrada en tiempo polinomial en una Máquina de Turing no determinista. Debido a que tal máquina no existe, prácticamente quiere decir que puede ser escrito un algoritmo exponencial para un problema NP, nada se asegura si existe o no un algoritmo polinomial para resolver todos los demás problemas NP. Finalmente, la clase NP-duros puede entenderse como la clase de problemas que son NP-completos o duros. Los problemas NP-duros tienen el mismo trato que los problemas NP-completo pero no necesariamente pertenecen a la clase NP, es decir la clase NP-dura incluye además problemas para los cuales no se puede proporcionar ningún algoritmo.

Las heurísticas están definidas por un conjunto de transformaciones permisibles de una solución actual, dígase Q , a una nueva solución (generalmente no probada), Q^* . El espacio de todas las soluciones posibles para un problema determinado se conoce como el *espacio de búsqueda*, S , y el espacio de soluciones que se pueden obtener por una simple transformación de la solución actual, Q , se conoce como el *espacio de vecindad*, N . (De Smith, Goodchild y Longley, 2006). Entre las heurísticas más conocidas se encuentran:

- a) Aleación Simulada: éste algoritmo usa una solución similar a escalamiento de colina, (Aydin y Fogarty, 2004), pero en ocasiones acepta soluciones que son peores a la actual. La probabilidad de tal aceptación decrece con el tiempo.
- b) Búsqueda Tabú: ésta búsqueda usa estructuras de memoria para evitar las soluciones óptimas locales. El problema de aleación simulada es que repite la búsqueda en soluciones ya exploradas. La búsqueda de tabú prohíbe la repetición de movimientos ya hechos anteriormente.
- c) Inteligencia de Enjambre: Es una técnica de inteligencia artificial, basada en el estudio de un comportamiento colectivo descentralizado, auto-organizado. Dos de los tipos más exitosos de ésta solución son Optimización de Colonia de hormiga (OCH) desarrollado por Dorigo y Stützle (2004), y Optimización de Enjambre de Partículas (PSO) desarrollado por Kennedy y Eberhart (2001). En OCH las hormigas artificiales construyen soluciones al moverse en el grafo del problema y cambian el ambiente de tal forma que las hormigas subsiguientes puedan encontrar mejores soluciones. PSO trata con problemas en la cual una solución buena puede ser representada como un punto o superficie de espacio n-dimensional. La ventaja principal de las técnicas de enjambre es que ellos son sorprendentemente resistentes al problema de óptimo local.
- d) Algoritmos evolutivos: éstos son exitosos abordando la convergencia prematura al considerar un número de soluciones simultáneamente. Producen aproximaciones graduales a la óptima aplicando el principio de sobrevivencia de un conjunto de soluciones potenciales.
- e) Las redes neuronales: están inspiradas en los sistemas neuronales biológicos. Se componen de unidades llamadas neuronas, y las interconexiones entre

ellas. Luego de un entrenamiento especial sobre algún determinado conjunto de datos, las redes neuronales pueden hacer predicciones para los casos que no están en el conjunto de entrenamiento. En la práctica, las Redes Neuronales no siempre funcionan bien porque presentan problemas de subentrenamiento y sobreentrenamiento (Kröse y Smagt, 1996). Estos problemas se correlacionan con la exactitud de la predicción. El problema de la convergencia prematura es también crítico para Redes Neuronales.

Algoritmos Metaheurísticos

Una metaheurística es un conjunto de conceptos algorítmicos que pueden ser usados para definir métodos heurísticos aplicables a una variedad de problemas. Puede ser vista como un método heurístico de propósito general diseñado para guiar una heurística acondicionada a un problema específico. Una metaheurística es un framework algorítmico general que puede ser aplicado a distintos problemas de optimización sin necesidad de hacerle muchas modificaciones.

En general, un algoritmo de optimización es un método numérico o algoritmo capaz de encontrar un valor (x) tal que $f(x)$ es tan pequeño (o tan grande) como sea posible, para una función f dada, posiblemente con algunas restricciones en x , donde x puede ser un vector escalar o vector de valores continuos o discretos.

Los algoritmos de optimización que usan metaheurísticas son denominados Algoritmos de Optimización Metaheurísticos. En dichos algoritmos existe un balance entre la calidad de una solución y la velocidad de procesamiento de tal solución.

Los Algoritmos Metaheurísticos, tal como se mencionó anteriormente, usan los métodos heurísticos estudiados para ser aplicados a distintos problemas. Como ejemplos se tienen: aleación simulada (Kirkpatrick, Gelatt y Vecchi, 1983), Búsqueda

Tabú (Glover y Laguna, 1997), Búsqueda Local Iterada (Lourenc ,Martin, y Stützle, 2002), Computación evolutiva (Holland, 1975; Goldberg, 1989), y Optimización de Colonia de Hormiga (Dorigo y Stützle, 2004).

Para la realización de pruebas de procedimientos de optimización combinatoria en general y en la heurística en particular, se utiliza a menudo el problema del comerciante viajero (conocido por sus siglas en inglés como TSP, derivado de *Traveling Salesman Problem*), como punto de referencia para medir la calidad y el rendimiento. Esto se debe a que el problema es fácilmente determinado y conocido por ser extremadamente difícil de resolver.

El Problema del Viajero Comerciante es uno de los problemas más intensamente estudiado en matemática computacional. Una de las aplicaciones actuales es la búsqueda de una ruta óptima en un mapa de Google Maps (Cook, 2010). En esencia, el TSP consiste en hacer una gira por todos los vértices de un conjunto dado (que pueden ser por ejemplo, las ciudades de un país, los nodos en una placa de circuito VLSI), y volviendo al punto de partida, de modo que la longitud total de la excursión o el costo sea mínimo. La medida de distancia utilizada debe ser una métrica o semi-métrica, es decir, la asimetría puede ser permitida, pero las distancias deben ser positivas y satisfacer la desigualdad del triángulo. Si hay N ciudades hay $N!$ posibilidades de visitas, por lo que el espacio de búsqueda es $O(N!)$, un valor que de forma rápida se hace inmensamente grande (esto reducirá ligeramente a $N! / 2$ si las distancias son simétricas y $(N-1)! / 2$ si la posición de la primera la ciudad está predefinida). (Smith, Goodchild y Longley, 2006)

Algoritmos Evolutivos

Éstos son métodos que usan ideas de evolución biológica, tales como reproducción, recombinación y mutación para buscar la solución a un problema de optimización. Ellos aplican el principio de sobrevivencia de un conjunto de soluciones potenciales para producir aproximaciones graduales a la óptima. Se crea un nuevo conjunto de aproximaciones mediante el proceso de selección de individuos de acuerdo a su función objetivo, la cual es llamada función adaptación (conocida como *fitness* en inglés) y usa operadores inspirados en procesos genéticos. Éstos procesos conllevan a una evolución de la población de los individuos que están mejor ajustados a su ambiente que sus ancestros.

El ciclo principal de los algoritmos evolutivos contiene los siguientes pasos:

1. Inicializar y evaluar la población inicial.
2. Llevar a cabo una selección competitiva.
3. Aplicar operadores genéticos para generar nuevas soluciones.
4. Evaluar soluciones en la población.
5. Iniciar de nuevo desde el punto 2 y repetir hasta que se satisfaga algún criterio de convergencia.

Los más exitosos entre los algoritmos evolutivos son los Algoritmos Genéticos (AGs). Éstos fueron investigados por John Holland en 1975 y demuestran una excepcional eficacia. Los AGs se basan en el hecho en que la mutación mejora al individuo muy rara vez y, por tanto se basan principalmente en la aplicación de los operadores de recombinación. Buscan soluciones a los problemas mediante series de

números, por lo general en binario. El área de aplicación prevalente en algoritmos genéticos es la optimización de problemas que requieren gran escala de recursos de cómputo y alto rendimiento. Por ejemplo, el problema de la asignación efectiva de recursos en relación con la Cubierta Plana de Acceso múltiple (conocida en inglés como Plane Cover Multiple Access PCMA) ha sido examinado por Blair, Polyzos y Zorzi en 2001. Su objetivo es maximizar la capacidad posible de conmutación de paquetes en redes inalámbricas celulares. El punto principal a considerar en la asignación de recursos, es reducir al mínimo el número de unidades de ancho de banda (UB) que deben ser asignados. El problema se ha comprobado que es de la clase NP-duros. Como resultado, se indica que el algoritmo genético puede mejorar la utilización de la capacidad del sistema.

Metaheurística de Optimización de Colonia de Hormigas

La Metaheurística de optimización de colonia de hormigas es un método probabilístico basado en el comportamiento natural de las colonias de hormigas reales. En la misma manera en que las hormigas van en búsqueda de una trayectoria desde su hormiguero hasta el alimento, el algoritmo de colonia de hormigas usa una colonia de hormigas artificiales en búsqueda de la mejor solución a un problema de optimización.

El factor clave de operación de estos algoritmos se fundamenta en la comunicación indirecta entre unos agentes de software, simulando el comportamiento de las hormigas. Esa forma de comunicación se conoce como estimergia en la cual las hormigas realizan modificaciones del ambiente obrando de manera cooperativa. Esto se lleva a cabo por una hormiga en el camino de vuelta al hormiguero luego de encontrar alimento, dejando un rastro de una pequeña cantidad de feromona en su trayectoria capaz de ser detectada por las demás hormigas. Las hormigas que

consigan el rastro de feromona depositado por la hormiga anterior, seguirán ese mismo camino para ir hasta la comida y volver, dejando a su vez un rastro nuevo de feromona por la misma trayectoria, aumentando la probabilidad de que las siguientes hormigas sigan el mismo camino hasta el alimento.

Debido a que varias hormigas pueden dejar su rastro de feromona por sus trayectorias individuales, esto podría ser un problema debido a que cada camino tendría la misma probabilidad, pero gracias a que las feromonas se evaporan con el tiempo, los caminos más largos tendrán niveles de feromona menores a los caminos más cortos permitiendo entonces que las hormigas tomen el camino más corto al alimento. De esa misma forma emergen las soluciones a problemas de optimización por los agentes de software al operar cooperativamente entre ellos (Dorigo y Stutzle, 2004). A continuación se presenta una caracterización formal de la representación que usan las hormigas y las políticas que implementan:

Consideremos el problema de minimización (S, M, Ω) , donde S es el conjunto de soluciones candidatas, f es la función objetivo que asigna un valor de función objetivo (costo) $f(s, t)$ para cada solución candidata $s \in S$, y $\Omega(t)$ es un conjunto de restricciones. El parámetro t indica que la función objetivo y las restricciones pueden ser dependientes del tiempo.

El objetivo es encontrar una posible solución óptima global s^* , es decir, una solución de costo mínimo factible para el problema de minimización. El problema de optimización combinatoria (S, M, Ω) es asignado a un problema que se caracteriza por:

- Un conjunto finito $C = \{c_1, c_2, \dots, c_N\}$ de componentes es dado, donde N_C es el número de componentes.
- Los estados del problema se definen en términos de secuencias de $x = \langle c_i, c_j, \dots, c_h, \dots \rangle$ de longitud finita sobre los elementos de C . El conjunto de todos los posibles estados se denota por X . La longitud de una secuencia x , es decir, el número de componentes de la secuencia, se expresa por $|x|$. La longitud máxima de una secuencia está limitada por una constante positiva $n < +\infty$.
- El conjunto de soluciones (candidatas) S es un subconjunto de X (es decir, $S \subseteq X$).
- Un conjunto de estados posibles de \bar{X} , con $\bar{X} \subseteq X$, que se define a través de una prueba dependiente del problema que comprueba que no es imposible completar una secuencia $x \in \bar{X}$ en una solución satisfaciendo las limitaciones Ω . Tenga en cuenta que según esta definición, la viabilidad de un estado $x \in \bar{X}$ debe ser interpretado en un sentido *débil*. De hecho, no es garantía de que existe una finalización s de x tal que $s \in \bar{X}$.
- Un conjunto no vacío S^* de soluciones óptimas, con $S^* \subseteq \bar{X}$ y $S^* \in S$.
- Un costo $g(s, t) \equiv f(s, t)$ se asocia a cada solución candidata $s \in S$. En la mayoría de los casos $g(s, t) = f(s, t)|$, $\forall s \in \tilde{S}$, donde $\tilde{S} \subseteq S$ es el conjunto de soluciones factibles candidatas, obtenidos de S a través de las restricciones $\Omega(t)$.

Dada ésta formulación, las hormigas artificiales construyen soluciones mediante la realización de caminos aleatorios por la gráfica completamente conectada $G_c = (C, L)$ cuyos nodos son los componentes C , y el conjunto L conecta completamente los componentes C . El grafo G_c se llama grafo de construcción y los elementos de L se denominan conexiones.

Las restricciones del problema $\Omega(t)$ están implementadas en la política seguida por las hormigas artificiales. La elección de implementar las restricciones en la política de construcción de las hormigas artificiales permite un cierto grado de flexibilidad. De hecho, dependiendo del problema de optimización combinatoria considerado, puede ser más razonable aplicar las restricciones de una manera dura, lo que permite a la hormigas construir sólo soluciones factibles, o de una manera suave, en cuyo caso las hormigas pueden crear soluciones infactibles (es decir, las soluciones de candidatos en $S \setminus \tilde{S}$) que pueden verse perjudicadas en función de su grado de infactibilidad.

Comportamiento de las hormigas

Como mencionó, en algoritmos OCH, las hormigas artificiales son procedimientos constructivos estocásticos que construyen soluciones al moverse a través del grafo de construcción $G_c = (C, L)$. Los componentes $c_i \in C$ y las conexiones $l_{ij} \in L$ pueden tener asociada un *rastro de feromona* τ (τ_i si está asociada con componentes, τ_{ij} si está asociada con conexiones), y el valor de heurística η (η_i y η_{ij} , respectivamente). El rastro de feromona codifica una memoria sobre el proceso entero de búsqueda y es actualizado por las hormigas. El valor heurístico, frecuentemente llamado información heurística, representa una información a priori acerca de la instancia del problema o información de corrida

dada por una fuente distinta a las hormigas. En muchos casos η es el costo (o un estimado del costo), necesario para agregar la conexión a la solución en construcción. Estos valores son usados por la regla heurística de las hormigas para tomar decisiones probabilísticas sobre cómo moverse en el grafo. Más concretamente, cada hormiga de la colonia (representada como hormiga k) tiene las siguientes propiedades:

- Se aprovecha el grafo de construcción $G_c = (C, L)$ para buscar soluciones óptimas $s^* \in S^*$
- Tiene una memoria M^k , donde k es cada hormiga de la colonia, que puede ser usada para almacenar información acerca del camino que siguió hasta ahora. La memoria puede ser utilizada para (1) construir soluciones factibles (es decir, aplicar restricciones Ω), (2) calcular los valores heurísticos η ; (3) evaluar la solución encontrada, y (4) volver a trazar el camino hacia atrás.
- Tiene un estado inicial x_s^k y una o más condiciones de terminación e^k . Por lo general, el estado inicial se expresa como una secuencia vacía o como una secuencia de longitud unidad, es decir, una secuencia de piezas individuales.
- Cuando en el estado $x_r = \langle x_{r-1}, i \rangle$, si no se satisface ninguna condición de terminación, se mueve a un nodo en su vecindad $N^k(x_r)$, es decir, a un estado $\langle x_r, j \rangle \in X$. Si se satisface al menos una de las condiciones de terminación e^k , entonces la hormiga se detiene. Cuando una hormiga construye una solución candidata, los movimientos a los estados infactibles están prohibidos en la mayoría de las aplicaciones, ya sea a través del uso de la memoria de la hormiga, o vía valores heurísticos η apropiadamente definidos para tal efecto.

- Se selecciona un movimiento mediante la aplicación de una regla de decisión probabilística. La decisión probabilística es una función de (1) rastros de feromona disponibles a nivel local y valores heurísticos (es decir, rastros de feromona y valores heurísticos asociados con componentes y conexiones en la vecindad de la ubicación actual de la hormiga en el grafo G_c), (2) almacenar el estado actual en la memoria privada de la hormiga, y (3) las restricciones del problema.
- Cuando se agrega un componente c_j al estado actual, se puede actualizar la ruta de feromona τ asociados con él o con su conexión correspondiente.
- Una vez que se ha construido una solución, se puede recorrer el mismo camino hacia atrás y actualizar los rastros de feromona de los componentes utilizados.

Algoritmo de Metaheurística OCH

Es importante tener en cuenta que las hormigas actúan concurrentemente y de forma independiente y que si bien cada hormiga es lo suficientemente compleja como para encontrar una solución (probablemente pobre) al problema en cuestión, las soluciones de buena calidad pueden sólo emerger como resultado de la interacción colectiva entre las hormigas. Se trata de un proceso de aprendizaje distribuido en el que los agentes individuales, las hormigas, no son adaptables por sí mismas, sino, por el contrario, modifican de forma adaptativa la forma en que el problema es representado y percibido por otras hormigas.

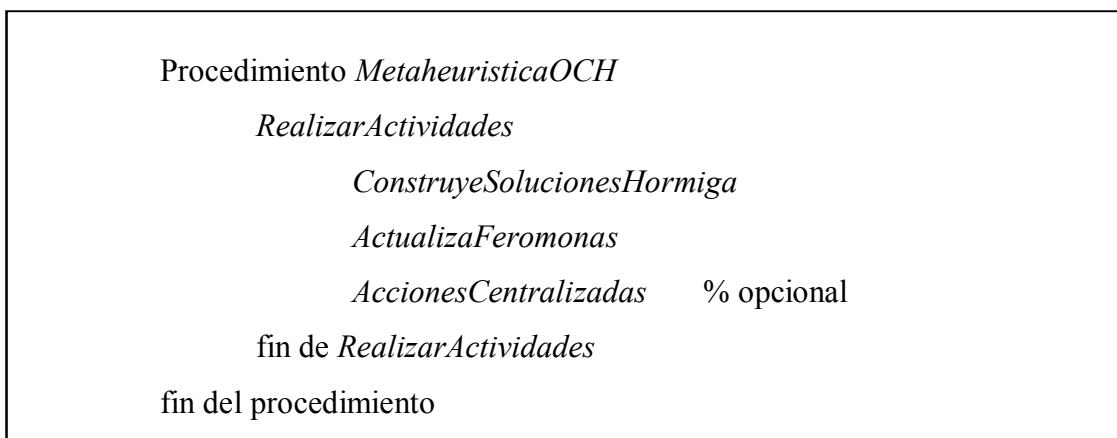
Un algoritmo OCH se puede plantear como la interacción de tres (3) procedimientos: *ConstruyeSolucionesHormiga*, *ActualizaFeromonas* y *AccionesCentralizadas*.

ConstruyeSolucionesHormiga gestiona una colonia de hormigas que al mismo tiempo y de forma asincrónica visitan los estados adyacentes de los problemas considerados moviéndose a través de nodos vecinos del problema del grafo de construcción G_c . Se mueven mediante la aplicación de una política de decisión local estocástica que hace uso de rastros de feromona y la información heurística. De esta manera, las hormigas gradualmente construyen soluciones al problema de optimización. Una vez que una hormiga ha construido una solución, o mientras la solución se está construyendo, la hormiga evalúa la solución (parcial) que será utilizada por el procedimiento *ActualizaFeromonas* para decidir cuánta feromona a depositar.

ActualizaFeromonas es el proceso mediante el cual los rastros de feromona son modificados. El valor de los rastros puede aumentar, mientras las hormigas

depositan feromona en los componentes o conexiones que utilizan, o decrementar, debido a la evaporación de feromona. Desde un punto de vista práctico, el depósito de feromonas aumenta la probabilidad de que los componentes y conexiones que fueron utilizados por muchas hormigas o que fueron utilizados al menos por una hormiga y que produjeron una muy buena solución, puedan ser usados por otras hormigas en el futuro. Por lo contrario, la evaporación de feromonas implementan una forma útil de olvidar: evita una convergencia demasiado rápida del algoritmo hacia una región sub-óptima, por lo tanto favoreciendo la exploración de nuevas áreas del espacio de búsqueda.

Por último, el procedimiento *AccionesCentralizadas* se utiliza para implementar acciones centralizadas que no pueden ser realizadas por las hormigas individuales. Como ejemplo, se puede observar el trayecto que han encontrado cada hormiga en la colonia y seleccionar unas pocas hormigas (por ejemplo, aquellas que construyeron las mejores soluciones) que luego permitirán depositar feromona adicional sobre la componentes y conexiones que utilizaban.



Cuadro 1: Algoritmo de la Metaheurística OCH.

El pseudo-código de la metaheurística OCH se describe en el cuadro 1. El procedimiento principal de la metaheurística OCH gestiona la programación de las tres componentes antes

discutidas de los algoritmos OCH a través de la construcción *RealizarActividades*: (1) gestión de la actividad de las hormigas, (2) actualización de feromona, y (3) acciones centralizadas. *RealizarActividades* no especifica cómo estas tres actividades se programan y sincronizan.

En otras palabras, no dice si deben ser ejecutadas de una manera totalmente paralela e independiente, o si algún tipo de sincronización es necesaria entre ellas. El diseñador es libre de especificar la forma en que estos tres procedimientos deben interactuar, teniendo en cuenta las características del problema considerado. En la actualidad existen disponibles numerosas implementaciones exitosas de la metaheurística OCH y han sido aplicadas a muchos problemas de optimización combinatoria.

CAPÍTULO III

MARCO METODOLÓGICO

Tipo de Investigación

El presente trabajo está diseñado bajo la modalidad de “Estudios de Proyectos”, de acuerdo al Manual para la presentación del trabajo conducente al grado académico de: Especialización, Maestría y Doctorado de la UCLA (2002), el cual lo define como:

Estudios de proyectos: Se entenderá por estudios de proyectos una proposición sustentada en un modelo viable para resolver un problema práctico planteado, tendente a satisfacer necesidades institucionales o sociales y pueden referirse a la formulación de políticas, programas, tecnología, métodos y procesos. (p. 5)

En este sentido, el estudio del proyecto partió de la construcción de una topología de red con una configuración aleatoria de distancias y capacidades de los sensores, como un modelo viable base que permitiría la orientación hacia la resolución de un problema práctico de eficiencia para el enrutamiento de datos en Redes de Sensores Inalámbricos, donde el punto más crítico está referido a la optimización de la energía de estos sensores. Por ello, el estudio se orienta buscar un algoritmo que satisfaga esta necesidad de eficiencia y ahorro, con el desarrollo de un modelo sustentado en la estructura de una colonia de hormigas artificiales que va en búsqueda del mejor camino a seguir por el paquete dentro de la red inalámbrica. Se considera como “mejor camino” aquel que consuma la menor energía y aumente la vida de la red. Por tanto, el problema se convierte en un problema de optimización.

Por otro lado, el trabajo se sustenta en su fase inicial, en un conjunto de trabajos previos tomados como antecedentes, como lo sugiere el Manual UCLA (ob. cit.): “Este tipo de estudio puede apoyarse tanto en la investigación de campo como en la investigación monográfica documental (p. 5). Es decir, se ha apoyado en investigaciones previas en donde se ha avanzado en diferentes vertientes que están vinculadas con un modelo de algoritmo propuesto, pero no han unificado estos elementos que darán fortaleza a este nuevo enfoque dirigido a la optimización de la disipación de la energía en los sensores y a la vez, mejorar la eficiencia en el enrutamiento en las redes que utilizan sensores inalámbricos.

Fases del Proyecto de Estudio

La investigación se consideró como un Proyecto de Estudio, como puede ser observado en investigaciones anteriores por autores como Dorigo, M. y Stützle (2004), Okdem, S. y Karaboga (2009), Yaofeng Wen, Yuquan Chen y Dahong Qian (2007), pues su propósito consistió en desarrollar un algoritmo que formara parte de un protocolo de red para el enrutamiento de paquetes en una red de sensores inalámbricos reales.

Factibilidad Técnica

Se requirieron recursos de hardware y software para la puesta en marcha de la investigación. Con respecto al software, fue factible el uso de "licencias libres" propuestas por el Gobierno Nacional Venezolano en el año 2001, ya que reduce los costos de implantación. Se sugiere entonces, un Sistema Operativo GNU/GPL por ejemplo Debían 10 o superior, licencia libre. Aunque podría instalarse en Sistemas Operativos de carácter propietario por ser un algoritmo transportable a cualquier plataforma (WINDOWS, SOLARIS, entre otros).

Por su naturaleza multiplataforma, el lenguaje de programación empleado para diseñar el algoritmo fue c++ utilizando Eclipse Europa y Cywin como compilador g++. Por otro lado, los requerimientos mínimos de hardware fueron: Procesador Pentium IV, 3 GB de RAM, 120 GB de disco duro, monitor a color, ratón, teclado, unidad de CD/RW.

Factibilidad Económica

Los recursos económicos necesarios para el desarrollo de la investigación estaban disponibles en la Unidad de Inteligencia Artificial del Decanato de Ciencias y Tecnología (UCLA). En dicha unidad se contó con un Servidor Pentium IV con capacidad similar a la descrita en el estudio de Factibilidad Técnica.

Procedimiento de la Investigación

En primer lugar, se diseñó un programa de simulación para la operación de la red de sensores inalámbricos. Los valores de distancia y capacidad de los sensores constituyen los datos de entrada para el Algoritmo basado en Colonia de Hormiga.

Para descubrir el “mejor” camino entre todos los posibles dentro de la red, se hizo necesario elaborar la función heurística de cálculo de la feromona de tal forma que se obtenga un valor alto cuando el consumo de energía era menor y bajo en caso contrario.

Una vez ejecutado el algoritmo de enrutamiento, se procedió a evaluar su rendimiento y convergencia hacia la solución mediante experimentos que utilizaban diferentes configuraciones de red, y a partir de allí, determinar los resultados y conclusiones.

CAPITULO IV

DISEÑO E INGENIERÍA DE LA PROPUESTA

Simulación de la Red

En la red de sensores inalámbricos considerada en la investigación, los nodos están dispuestos de tal forma que se interconectan unos con otros y a su vez se comunican con la estación central donde se procesa la totalidad de los datos capturados (Figura 5). Esta topología es conocida como topología de malla, la cual posee ventajas tales como: nodos visibles entre sí, escalabilidad que permite extender la red al agregar nuevos sensores y redundancia, la cual posibilita la elección de diferentes rutas para llegar al camino deseado (ver sección Bases Teóricas del Capítulo II). Así, al aplicar el algoritmo propuesto, se pueden obtener rutas alternas con bajos niveles de energía, creando una tolerancia a fallos permitida por la redundancia de la red.

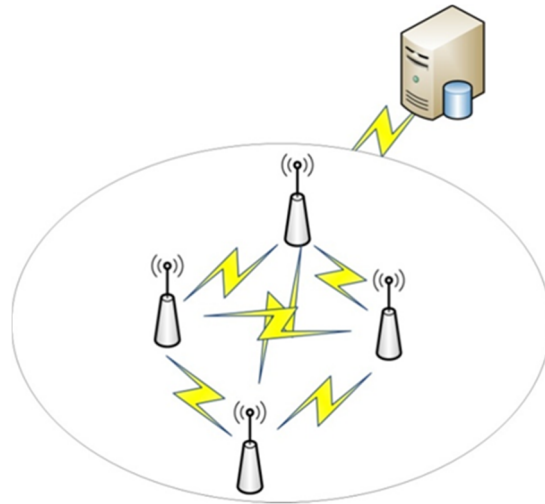


Figura 5: Red de Sensores Inalámbricos de la red en topología de Malla.

Cada cierto tiempo la estación central hace petición de fusión de datos. Para ello, envía un paquete especial llamado agente (Al-Karaki, 2005), el cual contiene la ruta de desplazamiento, recorre todos los nodos, fusiona la data y regresa a la estación central para su posterior procesamiento. Se asume que la red es homogénea, es decir que todos los nodos poseen las mismas características en cuanto a restricciones de energía y procesamiento y que son capaces de actuar tanto como nodos terminales como de enrutadores.

La estación central tiene conocimiento de toda la información de la red y las capacidades de cada sensor. A su vez, computa la ruta óptima para la fusión de datos utilizando el algoritmo de colonia de hormiga propuesto optimizando el consumo de energía total. Para ello, ubica a todas las hormigas en un nodo en particular llamado “nodo fuente” o “nodo inicial”, que en este caso es la estación base. A partir de allí, las hormigas inician su recorrido hasta que retornan al nodo inicial.

Matemáticamente, la red se traduce en un grafo completamente conexo $G_N = (N, A)$ cuyos nodos son los componentes de N y los enlaces de los sensores son el conjunto de arcos A . Para cada arco $(i,j) \in A$, se le asigna un valor de energía E_{ij} , el cual es representa el consumo de energía en el proceso de envío y recepción de datos entre el nodo i y el nodo j , con i y $j \in N$. Se desea encontrar una ruta de consumo mínimo de energía en el envío de un agente de software desde la estación base, que recorre cada uno de los nodos $n \in |N|$ del grafo G , fusiona data y regresa a la estación base (nodo fuente).

En la inicialización se cargan, vía archivo de texto, ciertos valores necesarios para la simulación, los cuales son los parámetros del algoritmo, los cuales se muestran en la tabla 1.

Tabla 1: Parámetros de la simulación del algoritmo

n	Número de nodos de la red de sensores inalámbricos.
m	Número de hormigas usadas para la construcción de rutas
nn	Cantidad de vecinos más cercanos que se calcula para cada nodo de la red.
α	Importancia de la tabla de feromonas en la selección de caminos.
β	Importancia de la heurística en la selección de caminos.
ρ	Tasa de evaporación global de feromona.
ξ	Tasa de evaporación local de feromona.
b	Tamaño en bits de los paquetes enviados
E_{elec}	Energía consumida en el circuito de recepción o transmisión
ϵ_{amp}	Energía consumida por el circuito amplificador de transmisión.

Adicionalmente se carga la información de la red por archivo de texto, es decir, la ubicación espacial de cada sensor representada como coordenadas (x, y) en un plano cartesiano, relacionadas con las posiciones reales de los sensores en el ambiente. Con las coordenadas de los sensores se calculan la matriz de distancia y la de vecinos más cercanos. A su vez se inicializan las matrices de rastros de feromonas y los valores de energía del sensor con valores aleatorios o seleccionados por el usuario de la aplicación.

De esta manera, la red de sensores inalámbricos está formada por un conjunto de n sensores en donde cada nodo se encuentra ubicado en un punto cartesiano (x,y) y contiene un valor de energía de la batería e dado en Joules. La representación de la red en el algoritmo contiene tres matrices principales las cuales son: los nodos, las distancias entre los nodos y la lista de vecinos más cercanos (Ver Cuadros 2 y 3).

```

//Representación del nodo
inicio
  real n           // identificación del nodo
  real x           // posición del nodo en el eje cartesiano x
  real y           // posición del nodo en el eje cartesiano y
  real energia    // valor actual de energía del nodo
fin

```

Cuadro 2. Estructura de datos para la representación de cada nodo de la red.

En el Cuadro 3 se observa una matriz de distancia *dist* de tamaño $n \times n$, ya que, como se detalla en la siguiente sección, es necesario conocer las distancias entre todos los sensores en la red. Los índices de la matriz coinciden con el identificador del nodo, es decir, el valor de *dist*[2][3] almacena la distancia entre el nodo de identificador 2 y el nodo con identificador 3.

```

//Representación del problema
inicio
  real dist[n][n] // matriz de distancia entre sensores
  entero lista_nn[n][nn] // matriz de vecinos mas cercanos de tamaño nn
  estructura nodo[n] // nodos de la red
  real energia_red // energía total promedio de la red
fin

```

Cuadro 3. Estructuras de datos para la representación del problema de enrutamiento en una RSI en la implementación del algoritmo OCH.

La matriz de vecinos más cercanos *lista_nn*, de tamaño $n \times nn$, contiene el identificador de todos los nodos vecinos de un nodo en particular, en un rango de distancia dado. Sea *dist*[*i*][*j*] la lista de las distancias del nodo *i* a todos los nodos *j*, con $j = 1, \dots, n$ donde $i \neq j$, la lista de vecinos más cercanos *lista_nn*[*i*][*z*] de un nodo *i* con $z = 1, \dots, nn$, contiene los primeros *nn* valores obtenidos al ordenar de mayor a

menor las distancias de la lista $dist[i][j]$. La lista *nodo* de tamaño n , almacena todos los nodos de la red con la estructura detallada en el Cuadro 2.

Distancia entre los nodos

Cada sensor en el proceso de envío de datos consume una cantidad de energía dada por las formulas (2.1) y (2.2), las cuales dependen de la distancia que existe entre el emisor y el receptor de la información. Dada las coordenadas (x, y) de los nodos sensores, la distancia $d(i, j)$ del nodo i al nodo j , está dada por la distancia Euclidiana entre los puntos (x_i, y_i) y (x_j, y_j) , calculada según la fórmula:

$$d(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (4.1)$$

Generalmente, las distancias entre los nodos son almacenadas en memoria como simples valores enteros. En este caso, se almacenan como valores reales debido a que las distancias entre los sensores pueden ser relativamente pequeñas en la vida real o contener valores decimales dependiendo de la unidad de medida empleada.

Representación de las Hormigas

Cada hormiga es un agente computacional que, partiendo del nodo fuente, se encarga de construir una ruta entre todos los sensores, aplicando una regla de actualización de feromona en cada arco (i, j) que traspasa. Para ello, debe almacenar la ruta parcial que ha construido, determinar el vecino de cada sensor, calcular y almacenar los valores de las soluciones que se generan.

```

//Representación de las Hormigas
estructura estructura_hormiga
inicio
    real energia_cons[n+1] //energía consumida por las hormigas en cada movimiento
    entero ruta[n+1] //memoria de las hormigas para almacenar las rutas parciales
    caracter visitado[n] //sensores visitados
fin
estructura_hormiga hormiga[m] //estructura de tipo estructura_hormiga

```

Cuadro 4. Estructura de datos para la representación de las hormigas

En el Cuadro 4, para cada hormiga, la lista de energía consumida *energia_cons* tiene tamaño $n+1$, ya que en cada posición i , guarda la energía consumida cuando parte de un nodo al siguiente. La posición adicional se usa para almacenar la energía que gasta cuando desde último regresa al nodo fuente con toda la información fusionada. Los índices de esta lista no coinciden con los identificadores de los nodos, por tanto, es necesario almacenar en la lista *ruta* los identificadores de los nodos que ha sido visitados por la hormiga, ordenados en el arreglo según su aparición. La estructura de datos *visitados* es inicializada en cero (0) y marca con el valor uno (1) el índice que coincide con el identificador del nodo que ya ha sido visitado. Esto con el fin de facilitar el proceso de búsqueda de la hormiga en la construcción de la ruta. La lista de hormigas *hormiga* tiene tamaño m , dado en el archivo de inicialización de parámetros del algoritmo.

Rastros de Feromona

Para cada conexión (i, j) se le asigna un valor τ_{ij} que representa el rastro de feromona asociado con dicha conexión. Por ello, se almacenan los rastros de feromona en la matriz “feromona” de tamaño n^2 , la cual es simétrica debido a que se asume que $\tau_{ij} = \tau_{ji}$, $\forall(i, j)$ (Ver Cuadro 5).

Información total de feromona y Heurística

En el proceso de construcción de la ruta, una hormiga en un sensor i elige al siguiente sensor j con una probabilidad de p_{ij} , según indica la fórmula metaheurística detallada mas adelante. Debido a que esta información es usada para cada hormiga, es necesario el uso de la matriz simétrica “total” (Cuadro 5), de tamaño n^2 para almacenar los valores resultantes de $[\tau_{ij}]^\alpha [\eta_{ij}]^\beta$ (Bases Teóricas, Capítulo II).

```
//Representación de la matriz de feromona y heurística
inicio
  real feromona[n][n]           // matriz de feromonas
  real total[n][n]             // total de información de feromona y heurística
fin
```

Cuadro 5. Estructuras de datos para la representación de la matriz de feromona y el total de información heurística.

Algoritmo de enrutamiento basado en Colonia de Hormigas.

En la presente sección se describen los pasos de implementación del Algoritmo de Colonia de Hormigas aplicado al problema de la Red de Sensores Inalámbricos, en el cual, la construcción de caminos, es un aspecto clave para la búsqueda de soluciones. La ruta para el envío de la data se construye aplicando el siguiente proceso: Primero se envía un paquete hormiga desde el nodo fuente. Luego se usan los valores de feromonas y heurísticas para construir de manera probabilística una ruta, agregando sensores que no se han visitado aún, fusionando data de cada nodo hasta que todos los sensores sean visitados para así culminar el recorrido hacia la estación base o nodo inicial. Cada vez que la hormiga usa un arco para moverse de un sensor a otro, actualiza los rastros de feromona, reduciéndola para incrementar la exploración de caminos alternos. Otro paso importante es la actualización global de

los rastros de feromona, en el cual se lleva a cabo el depósito y evaporación de feromona en los arcos de la ruta obtenida por la mejor hormiga.

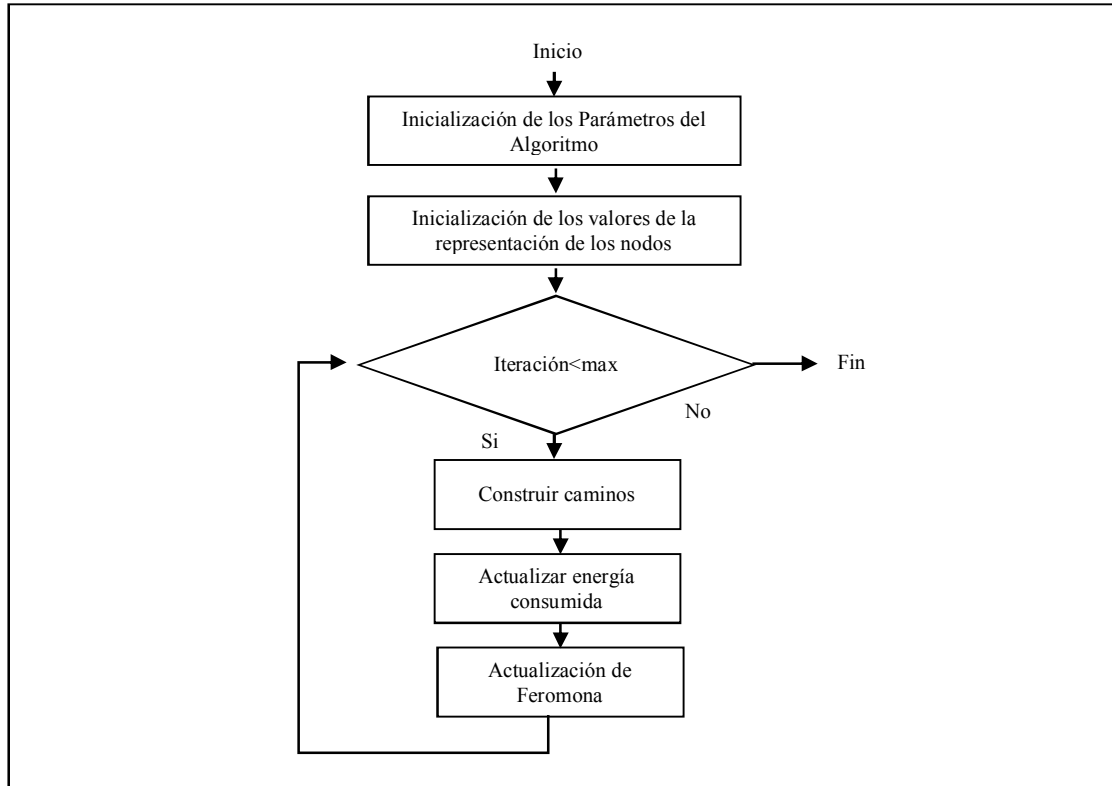


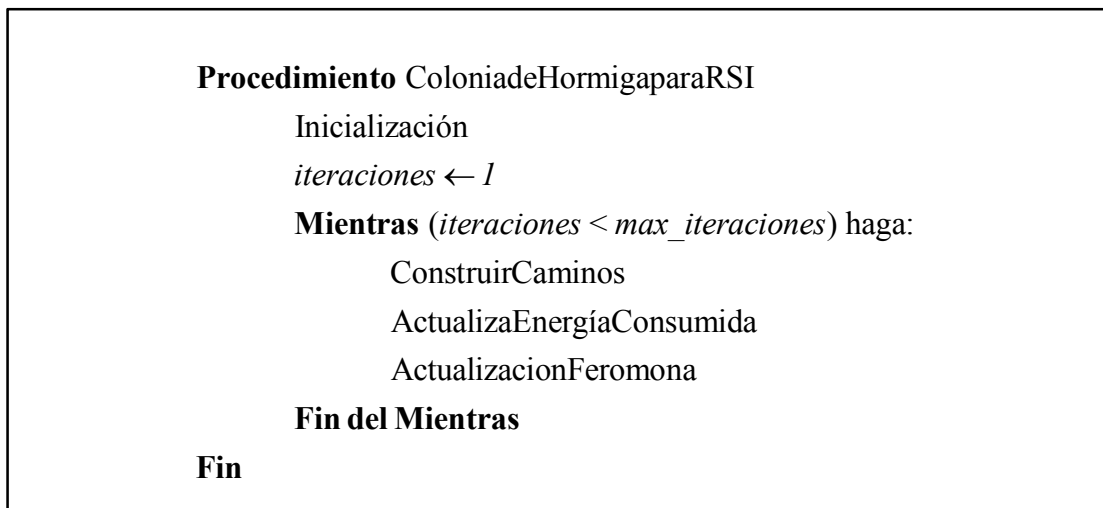
Figura 6. Diagrama de flujo del Algoritmo de Colonia de Hormigas aplicado al problema de Red de Sensores Inalámbricos.

El diagrama de flujo del algoritmo de Colonia de Hormigas, mostrado en la Figura 6, presenta tres principales tareas. La primera es el proceso llevado a cabo por las hormigas para construir rutas de comunicación entre sensores, incluida la actualización local de feromona. La siguiente es calcular el consumo de energía promedio en las rutas conseguidas por las hormigas en el proceso construcción de caminos el cual es usado para elegir la mejor hormiga de la iteración. Finalmente el algoritmo debe gestionar la actualización y evaporación de los rastros de feromona llevado a cabo por la mejor hormiga.

Adicionalmente y fuera del ciclo principal, se lleva a cabo la inicialización de parámetros y estructuras de datos del algoritmo y la información de la red con los parámetros de la simulación detallada al inicio del capítulo.

Procedimiento Principal del Algoritmo de Colonia de Hormigas

Como se mencionó anteriormente, las principales tareas del algoritmo de Colonia de Hormigas para las RSI son construir rutas de comunicación de los sensores, calcular el consumo de energía promedio en las rutas y gestionar la actualización de los rastros de feromona (Cuadro 6). Este proceso se repite mientras no se alcance el máximo de iteraciones.



Cuadro 6. Procedimiento Principal del OCH para la RSI

Construcción de los caminos

En esta fase, inicialmente se limpia la memoria de los caminos visitados por las hormigas. Luego, se ubica a las hormigas en el nodo inicial. Cada hormiga comienza una ruta desde dicho nodo, recorre todos los sensores y culmina su camino de regreso

al nodo fuente (estación base). El pseudocódigo del procedimiento *ConstruirCaminos* se muestra en el Cuadro 7.

```

Procedimiento ConstruirCaminos
  desde  $k = 1$  hasta  $m$ 
    desde  $i = 1$  hasta  $n$ 
       $hormiga[k].visitado[i] \leftarrow falso$ 
    fin - desde
  fin - desde
   $paso \leftarrow 1$ 
  desde  $k = 1$  hasta  $m$ 
     $r \leftarrow estacionbase$ 
     $hormiga[k].ruta[paso] \leftarrow r$ 
     $hormiga[k].visitado[r] \leftarrow verdadero$ 
  fin - desde
  mientras ( $paso < n$ )
     $paso \leftarrow paso + 1$ 
    desde  $k = 1$  hasta  $m$ 
      SeleccionaVecinoSiguiente( $k, paso$ )
      ActualizaciónLocalFeromona( $k, paso$ )
    fin - desde
  fin - mientras
   $paso \leftarrow n$ 
  desde  $k = 1$  hasta  $m$ 
     $hormiga[k].ruta[n+1] \leftarrow hormiga[k].ruta[r]$ 
    ActualizaciónLocalFeromona( $k, paso$ )
  fin - desde
fin - procedimiento

```

Cuadro 7. Procedimiento para la construcción de las rutas.

En cada paso, una hormiga situada en el nodo r , selecciona el siguiente nodo (VecinoSiguiente) usando la regla probabilística propuesta en la Meta-heurística:

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} & \text{si } j \in N_i^k \\ 0 & \text{en cualquier otro caso} \end{cases} \quad (4.6)$$

donde:

k : Es la hormiga situada en el nodo i .

p_{ij}^k : Es la probabilidad en la cual la hormiga k elige para moverse desde un nodo i a un nodo j .

τ_{ij} : Es la abreviación de la tabla “feromona” donde se almacena la cantidad de rastros de feromona en la conexión que hay entre los nodos i, j .

$\eta_{ij} = E_j$: Es una función heurística propuesta en la investigación.

α y β : Parámetros que controlan la importancia relativa de los rastros de feromona versus la visibilidad.

N_i^k : Vecindad factible o vecinos más cercanos de una hormiga k cuando se encuentra en un nodo i , es decir el conjunto de nodos que una hormiga k no ha visitado aun (la probabilidad de elegir un sensor fuera de N_i^k es 0).

La función heurística, entre los nodos i y j , viene dada por:

$$\eta_{ij} = E_j, \text{ dado que } E_j = \frac{1}{J - e_j}$$

donde, J es el nivel de energía inicial de los nodos y e_j es la energía actual del nodo j .

Así, la probabilidad de elegir un arco particular (i,j) se incrementa con el valor de información heurística η_{ij} y el valor de feromona asociado τ_{ij} . La probabilidad de selección es un balance entre visibilidad y la intensidad actual del rastro de feromona.

La visibilidad permite que los nodos con mayor energía sean elegidos con mayor probabilidad, mientras que los rastros de feromona indican que ha habido mayor tráfico por esa ruta, lo cual se convierte en una mejor opción.

Actualización local de los rastros de feromona

Cuando una hormiga se mueve al siguiente sensor, se realiza la actualización local de ratros de feromona, propuesto en Dorigo y Stützle (2004). En el procedimiento *ActualizaciónLocalFeromona(k, paso)*, la hormiga calcula la cantidad de feromona que depositará en su recorrido de la siguiente manera:

$$\tau_{ij} \leftarrow (1 - \xi)\tau_{ij} + \xi\tau_0 \quad (4.8)$$

El parámetro ξ , $0 < \xi < 1$, es la tasa de evaporación de feromona. Según los experimentos realizados en el trabajo de Dorigo y Stützle (2004), el valor de ξ sugerido es de 0,1. τ_0 es la cantidad de feromona inicial elegida como:

$$\tau_0 = \frac{1}{nD^{nm}} \quad (4.9)$$

donde D^{nm} : es la longitud de la ruta dada por la lista de los vecinos más cercano. El valor de D^{nm} es calculado a priori mediante la matriz de distancias entre los nodos dada previamente en la estructura de la representación de la red.

La implementación de la actualización local de feromonas es mostrada en el Cuadro 8, el cual es invocado por el procedimiento *ConstruirCaminos* mostrado en el Cuadro 7. En este procedimiento, cada vez que una hormiga usa un arco (i,j) , reduce su valor de feromona τ_{ij} , lo cual hace que su ruta sea menos deseable por las siguientes hormigas, provocando así un comportamiento emergente de exploración de rutas.

procedimiento ActualizacionLocalFeromona(k,paso)

$j \leftarrow hormiga[k].ruta[paso]$

$h \leftarrow hormiga[k].ruta[paso - 1]$

$feromona[j][h] = (1 - \xi) * feromona[j][h] + \xi * \tau_0$

$feromona[h][j] \leftarrow feromona[j][h]$

$total[j][h] \leftarrow \exp(feromona[j][h], \alpha) * \exp(1 / (J - instancia.nodo[j].energia), \beta)$

$total[h][j] \leftarrow total[j][h]$

fin - procedimiento

Cuadro 8. Procedimiento para la actualización local de feromona.

Actualización de energía consumida

En el cálculo de los costos de la red para el envío y recepción de data se tomó el modelo de gasto de energía propuesto por Heinzelman, et al. (2000), visto en la sección Consumo de energía de la red del capítulo II. Una vez construida la ruta de cada hormiga, se calcula su energía promedio consumida mediante la fórmula:

$$Ec = \text{Prom}(E) - \text{Prom}(R) \quad (4.7)$$

donde $\text{Prom}(E)$ es el promedio de los valores del vector E , el cual contiene los valores de energía existente en los nodos de la red, calculado por:

$$\text{Prom}(E) = \frac{\sum_{i=1}^n e_j}{n}$$

$\text{Prom}(R)$ es el promedio de los valores de energía restante de los nodos de la ruta, dado por:

$$\text{Prom}(R) = \frac{\sum_{j,i \in A^k} ((e_j - Etx_j) + (e_i - Erx_i))}{n}$$

En la fórmula, e_i y e_j representan las energías actuales de los nodos i , j , y Etx_i y Erx_j son las energías consumidas por cada nodo respectivamente, las cuales son calculadas mediante las formulas (2.1) y (2.2). A^k es el conjunto de las aristas de la ruta de la hormiga k .

El valor de la energía consumida es almacenado en cada hormiga por la variable *energía_cons* y se calcula en el procedimiento ActualizaEnergiaConsumida mostrado en el Cuadro 9, llamado a su vez, desde el procedimiento principal (Ver Cuadro 6).

```

procedimiento ActualizaEnergiaConsumida(k, paso)
  real prom_energia = 0.0;
  real energia = 0.0;
  para i = 1 hasta paso
    j = hormiga[k].ruta[i - 1];
    h = hormiga[k].ruta[i];
    instancia.nodo[j].energia -= Etx(j, k);
    instancia.nodo[h].energia -= Erx(h, k);
    prom_energia += (instanc.nodo[j].energia + instanc.nodo[h].energia);
  fin - para
  prom_energia = prom_energia / paso;
  hormiga[k].energia_cons = instancia.energia_red - prom_energia;
  return prom_energia;
fin - procedimiento

```

Cuadro 9. Procedimiento de actualización de consumo de energía de cada ruta.

El algoritmo descrito en el Cuadro 9 recibe los parámetros k , el cual representa el índice de la hormiga, y $paso$, que representa el tamaño de la ruta y coincide con el número total de nodos. En cada iteración del ciclo, se selecciona los nodos i y j , se calcula la energía restante y se actualizan las energías de dichos nodos. La energía existente de la red almacenada en la variable *instancia.energia_red*, es calculada en

el momento que se realiza la inicialización de los valores de energía de los sensores descrita previamente en la simulación de la red. Luego se promedian los valores obtenidos y finalmente se computa la energía consumida para la hormiga k . Cada vez que se calcula la energía consumida por cada hormiga, se fijan los valores de energía de los nodos a la configuración inicial de la red. Esto con el fin de evitar que la ruta de una hormiga influya en el consumo de energía de cualquier otra.

Actualización global de los rastros de feromona

Luego de la actualización de la energía consumida, se selecciona la mejor hormiga de la iteración actual, denominada *mejor_hormiga_iteración*, la cual es la que posee el menor valor de energía consumida en su ruta. Si la energía consumida de dicha hormiga es menor que cualquiera de las obtenidas en las iteraciones anteriores, entonces se designa como la mejor hormiga de todas las iteraciones y se le denomina mj . A ésta hormiga se le permite agregar feromona al final de cada iteración, la cual está dada por la siguiente ecuación:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}^{mj}, \quad \forall(i, j) \in T^{mj} \quad (4.10)$$

donde,

$$\Delta\tau_{ij}^{mj} = \frac{1}{Ec^{mj}} \quad (4.11)$$

la cual es la fórmula que se propone para la actualización del rastro de feromona, la cual es depositada por la mejor hormiga para que de manera cooperativa junto con las demás hormigas se consiga una ruta que tenga un consumo de energía cada vez menor.

El valor de Ec^{mj} es el consumo de energía generado por la mejor hormiga y es calculado tal como se muestra en la ecuación (4.7) en el procedimiento ActualizaEnergiaConsumida.

El parámetro ρ es la tasa de evaporación de feromona y debe estar entre 0.0 y 1.0 tal como se presenta en Dorigo y Stützle (2004). Este parámetro es usado para evitar la acumulación ilimitada de feromona y permite que el algoritmo pueda buscar nuevas soluciones explorando rutas no visitadas por las hormigas.

procedimiento ActualizacionFeromona
 ActualizacionGlobalFeromona(hormiga[mj])
fin - procedimiento

Cuadro 10. Procedimiento para la actualización de feromona.

La implementación de la actualización global de feromonas se muestra en el Cuadro 11, el cual es llamado por el procedimiento ActualizacionFeromona mostrado en el Cuadro 10. Su función es de reforzar la feromona en el camino conseguido por la mejor hormiga para aumentar la probabilidad de escoger esa ruta por las siguientes hormigas.

procedimiento ActualizacionGlobalFeromona(k)
 $\Delta\tau = 1 / \text{hormiga}[k].\text{energia_cons}$
para $i = 1$ **hasta** n
 $j \leftarrow \text{hormiga}[k].\text{ruta}[\text{paso}]$
 $h \leftarrow \text{hormiga}[k].\text{ruta}[\text{paso} + 1]$
 $\text{feromona}[j][h] = (1 - \rho) * \text{feromona}[j][h] + \rho * \Delta\tau$
 $\text{feromona}[h][j] \leftarrow \text{feromona}[j][h]$
 $\text{total}[j][h] \leftarrow \exp(\text{feromona}[j][h], \alpha) * \exp(1 / (J - \text{instancia.nodo}[j].\text{energia}), \beta)$
 $\text{total}[h][j] \leftarrow \text{total}[j][h]$
fin - para
fin - procedimiento

Cuadro 11. Procedimiento para la actualización global de feromona.

En líneas generales, los procedimientos mostrados anteriormente conforman la parte fundamental del algoritmo. Su implementación en el lenguaje de programación C++ puede ser visto en detalle como anexo del presente documento donde se presentan los procedimientos usados para la realización de los experimentos que se detallan en el siguiente capítulo.

CAPITULO V

EXPERIMENTOS Y RESULTADOS

En esta sección se presentan los resultados experimentales obtenidos por el algoritmo. En cada corrida del algoritmo, se realizó la inicialización de variables y parámetros requeridos en ciertos valores, los cuales se muestran en la tabla 2.

Tabla 2. Valores de los parámetros de la simulación del OCH

Número de vecinos más cercanos nn	15
Influencia relativa de la información heurística α	0,85
Influencia relativa de los rastros de feromona β	0,75
Factor de evaporación de feromona global ρ	0,5
Factor de evaporación de feromona local ξ	0,1
Tamaño de la data transmitida b	2000
Energía consumida en el circuito E_{elec}	50 nj/bit
Energía consumida por el circuito amplificador ϵ_{amp}	100 $pj/bit/m^2$

Se estableció el número de vecinos más cercanos en $nn = 15$, ya que comúnmente el valor de nn está entre 15 y 40 (Dorigo y Stützle 2004). Debido a que la función de este valor es disminuir la complejidad al elegir el siguiente nodo, su valor es fijado al mínimo para así aumentar la velocidad de ejecución del algoritmo. El valor de α fue escogido en 0,85 y β en 0,75 para dar mayor peso a la tabla de feromonas de las hormigas que a la heurística en la construcción de las rutas (Dorigo et. al., 1992; 1996). Adicionalmente el valor de ρ , discutido en el capítulo anterior, debe estar entre 0,0 y 1,0 pero su valor se escogió en $\rho = 0,5$ tal como es sugerido en

experimentos de Dorigo (1992, 1996). Por otra parte, el valor de ξ es de 0,1 tal como se sugiere en el capítulo anterior.

El modelo de radio usado está basado en el trabajo de Heinzelman et. al. (2000) para la cantidad de b bits de datos transmitidos por los nodos y para los valores de E_{elec} y ϵ_{amp} , lo cual corresponde, según lo mostrado en sus investigaciones, a un modelo ligeramente mejor en el estado de arte actual para el diseño de radio, por ejemplo, la tecnología Bluetooth con radios que operan a 115 nj/bit/m².

Los sensores fueron colocados de forma aleatoria en un espacio bidimensional, similar al trabajo de Heinzelman et. al. (2000). En la primera fase de experimentación se usaron 100 nodos colocados en un área de 50x50 m. A cada nodo se le dio inicialmente 0,5J de energía. La disposición de los sensores se muestra en la Figura 7.

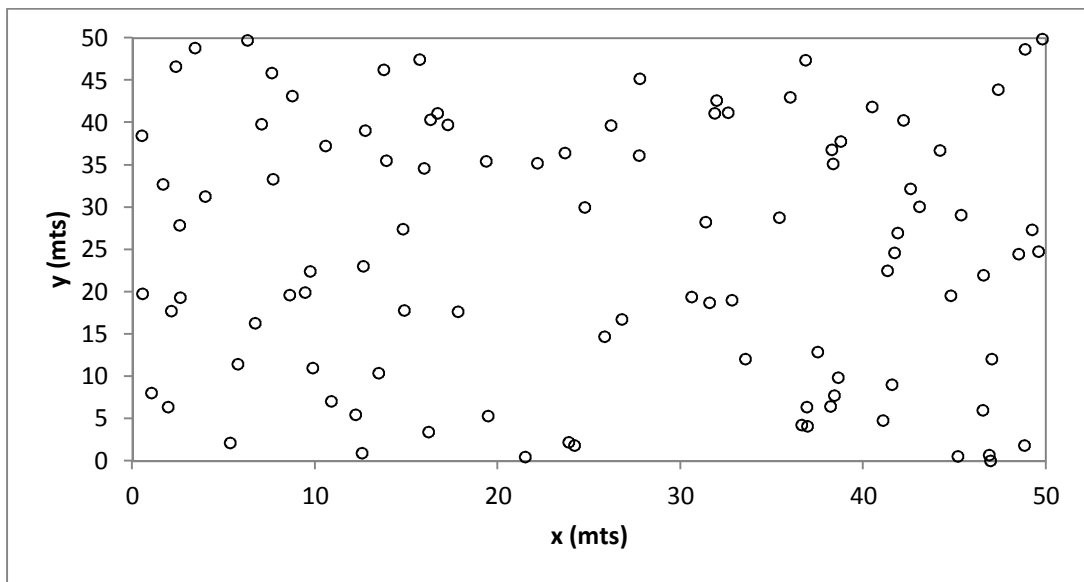


Figura 7. Distribución espacial de los nodos en la red aleatoria de 100 sensores.

Las métricas usadas para medir el rendimiento del algoritmo, en términos de la optimización del consumo de energía, vienen dadas por el consumo de energía total

en la ruta recorrida por las mejores hormigas y medidas estadísticas tales como, media y desviación estándar (Walpole y Myers, 1992), de los consumos de energía alcanzados por todas las hormigas. A continuación, se detallan los resultados promedios de estas medidas a lo largo de las iteraciones, en cada experimento.

Se realizaron 100 experimentos de 1000 iteraciones cada uno desplegando $m=25$ hormigas, y se computaron los consumos de energía alcanzados por el camino de cada mejor hormiga. Los valores de energía de cada iteración se promediaron entre el número de experimentos y el resultado se muestra en la Figura 8.

Se puede observar que el consumo de energía total de la red desciende rápidamente a valores cercanos al mínimo obtenido en aproximadamente 50 iteraciones. A partir de allí, la función se estabiliza alrededor de la media. El valor del consumo de energía comienza con valores pequeños debido a que el valor de feromona inicial $\tau_0 = 2.1551724138 \times 10^{-5}$ beneficia la convergencia del algoritmo.

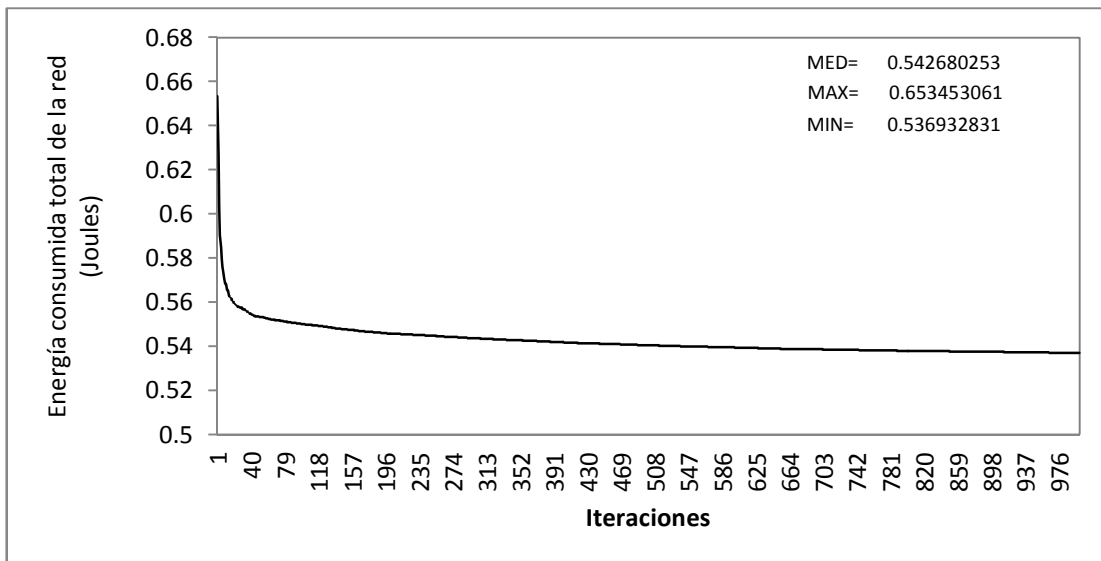


Figura 8. Energía consumida total en la ruta obtenida por la mejor hormiga.

Por otro lado, para verificar que todas las hormigas tienen el mismo comportamiento promedio que se observó en la gráfica anterior, se procedió a calcular la media de los valores de consumo de energía de todas las soluciones. Los resultados se muestran en la Figura 9, donde se puede observar un comportamiento hacia la convergencia similar al mostrado en la gráfica anterior.

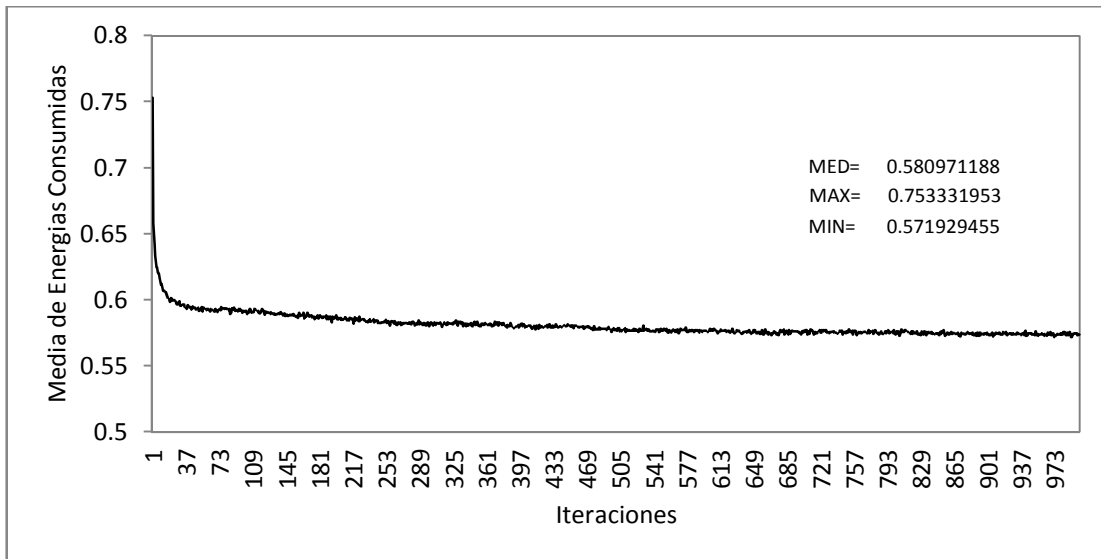


Figura 9. Media de los valores de energía de las rutas de las hormigas

Mediante la desviación estándar se puede medir la dispersión de los valores de energía con respecto a la media. Los resultados se muestran en la Figura 10, donde se observa que los valores oscilan entre un máximo de 0.0517 y un mínimo de 0.0270 con tendencia descendente. Estas pequeñas variaciones indican alta concentración de los datos alrededor de la media.

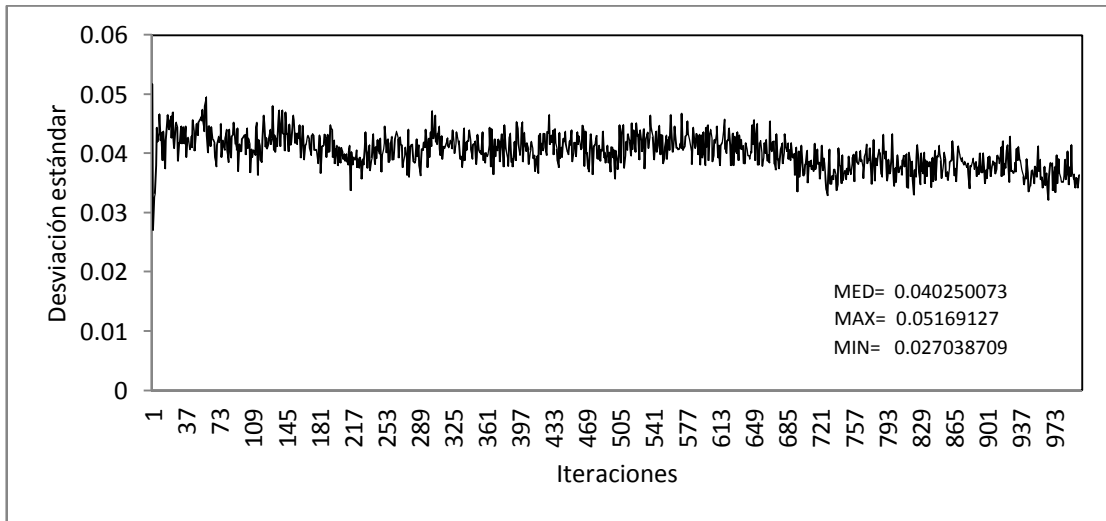


Figura 10. Desviación estandar de los valores de energía de las rutas de las hormigas

Para verificar si todas las hormigas siguen la misma ruta y construyen la misma solución, se procedió a verificar la convergencia del algoritmo. Una métrica usada para dicha convergencia, es mediante la formula de distancia $dist(R,R')$ entre dos rutas dadas R y R' (Dorigo y Stützle, 2004), la cual consiste en contar el número de arcos que estan en la ruta R que no están en la ruta R' . El resultado obtenido por esta métrica se muestra en la Figura 11.

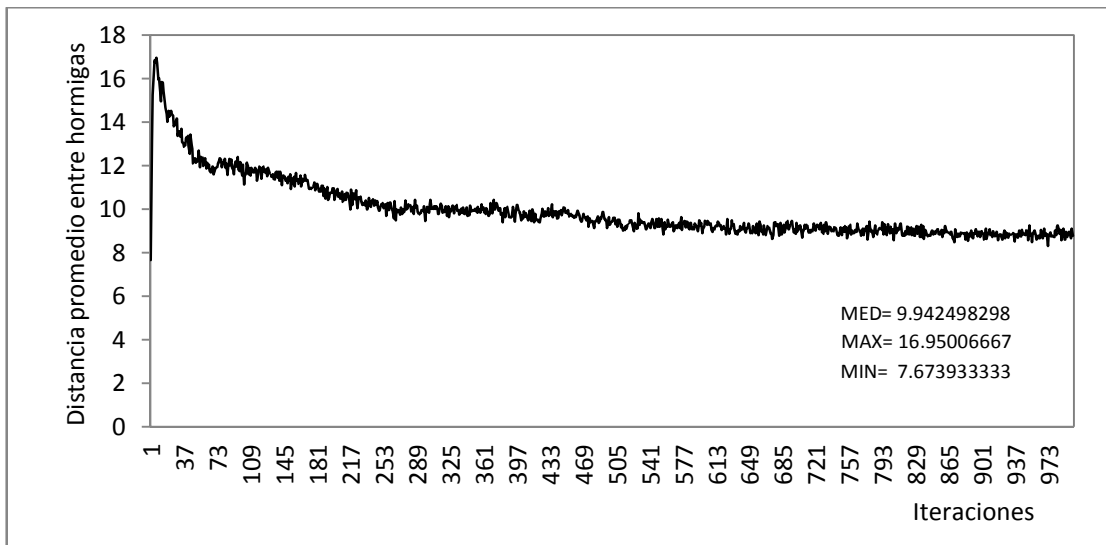


Figura 11. Distancia entre las hormigas con respecto a sus rutas

El decremento de la distancia entre rutas, observada en la Figura 11, indica preferencia de las hormigas por una ruta en particular, la cual, de acuerdo a los resultados obtenidos en la sección anterior, es similar a la ruta de la mejor hormiga. Sin embargo, su comportamiento no tiende a cero ya que no todas las hormigas escojen una misma ruta, lo cual se debe a la exploración continua de las hormigas hacia otras rutas para evitar una convergencia prematura.

De la misma manera, se usó la métrica para medir la distancia entre las rutas de las mejores hormigas. Para ello, se tomó la ruta de la mejor hormiga en una iteración y se comparó con la iteración anterior. En la Figura 12, se puede apreciar que la distancia tiende a cero a medida que transcurren las iteraciones, lo cual significa que las mejores hormigas tienden a escoger una misma ruta.

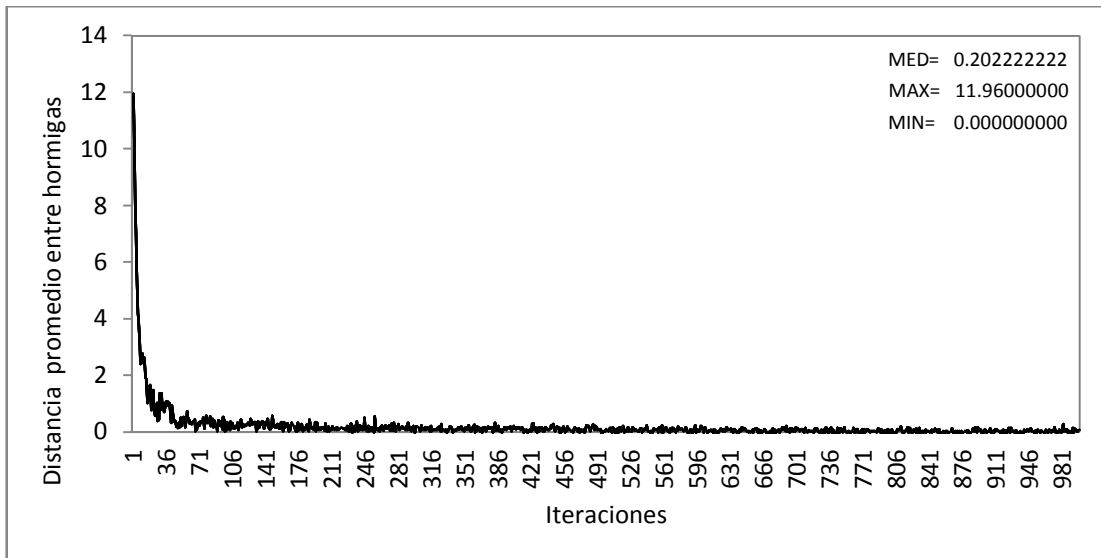


Figura 12. Distancia con respecto a la ruta entre las mejores hormigas.

Otra forma comúnmente usada para medir la convergencia del algoritmo, consiste en la aplicación de la siguiente fórmula (Dorigo y Stützle, 2004):

$$Conv = \frac{\sum_{\tau_{ij} \in T} \min\{\tau_{\max} - \tau_{ij}, \tau_{ij} - \tau_{\min}\}}{n^2} \quad (5.1)$$

donde τ_{\max} y τ_{\min} son los valores máximo y mínimo que se obtienen al buscar iteradamente en la tabla de feromonas. Si los valores τ_{ij} están muy cercanos a τ_{\min} ó τ_{\max} , indica una clara preferencia de las hormigas por aristas muy similares a τ_{\max} , en contraste a otras poco transitadas cercanas a τ_{\min} . Por tanto, el mínimo de las diferencias se hace muy pequeño y el resultado tiende a cero. Si por el contrario, la tabla de feromonas refleja valores τ_{ij} muy distantes de τ_{\min} ó τ_{\max} , se obtendría un resultado alejado de cero, ya que las hormigas no han marcado preferencias en la ruta de visita. En consecuencia, si el resultado de ésta fórmula tiende a cero a medida que transcurren las iteraciones, significa que el algoritmo converge a una solución.

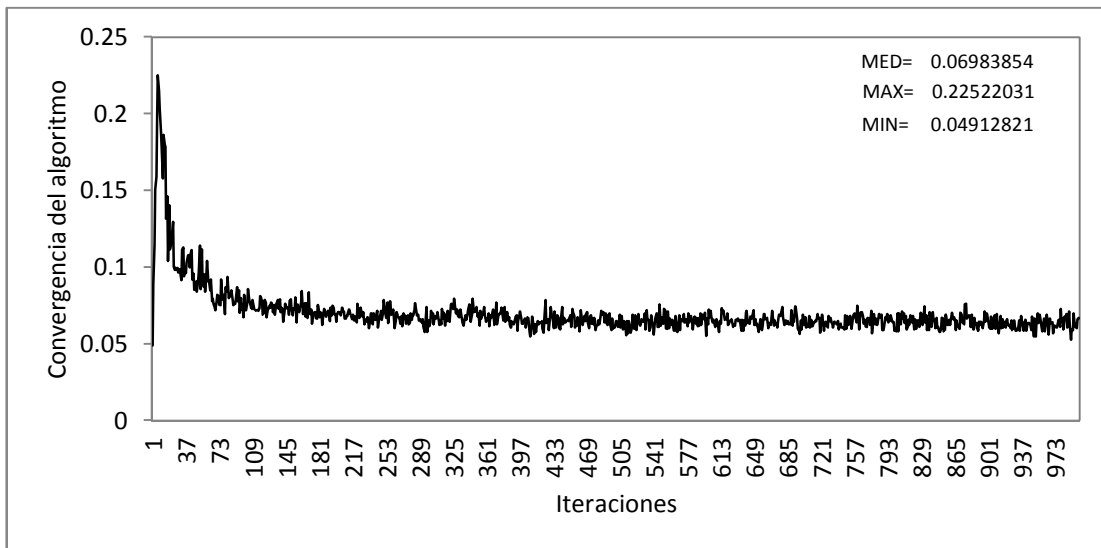


Figura 13. Convergencia del algoritmo

En la Figura 13, se muestran los resultados dados por la ecuación 5.1. Paulatinamente, la métrica *Conv* se decrementa en valores cercanos a cero, lo cual indica que el algoritmo tiene un comportamiento hacia la convergencia. Esto significa

que el algoritmo es capaz de encontrar rutas con consumos de energía alrededor de un mínimo.

De forma general, las mejores soluciones obtenidas por el algoritmo en las diferentes iteraciones reducen el consumo de energía convergiendo alrededor de un mismo valor. La distancia entre estas soluciones son muy cercanas a cero, lo cual indica que el algoritmo es capaz de encontrar una ruta con bajo consumo de energía.

Cabe destacar que la implementación del algoritmo de colonia de hormiga propuesto en el presente trabajo, difiere de otros trabajos tal como el de Camilo et. al. (2006), por lo cual no es posible establecer comparaciones. En su configuración de red, los sensores pueden estar configurados en malla, o en una topología híbrida donde los nodos cabecera pueden estar en cualquier localidad desconocida inicialmente por el algoritmo. Cada sensor envía su información independiente a la estación base, usando sensores intermedios en una ruta determinada por dos clases de hormigas. Una de ellas es la *hormiga directa* que van en el sentido de la ruta hacia el destino o estación base. Una vez que la *hormiga directa* llega al destino, se envía una *hormiga reversa* la cual tiene la misión de actualizar el rastro de feromona usado para alcanzar el destino y lo almacena en su memoria. Dicha actualización de feromona, depende del número de nodos que la *hormiga directa* visita. Finalmente el algoritmo fue implementado como parte de un protocolo de enrutamiento, tomando en cuenta la movilidad en tiempo real.

En el presente trabajo, la configuración de la red es determinante en la forma en que se transmiten los datos, ya que las hormigas realizan sus recorridos en toda la red en un grafo totalmente conexo, con una configuración de topología malla. Adicionalmente, el nodo inicial coincide con el último al que llegan los datos, el cual es la estación base. Tanto la información heurística utilizada por las hormigas como

la actualización de las feromonas son acondicionadas al problema específico. El proceso de actualización de feromona se realiza de dos formas, localmente por cada una de las hormigas y globalmente por la mejor hormiga.

Por tanto, se provee una solución distinta en la búsqueda de rutas de bajo consumo global de energía en una RSI configurada en topología tipo malla, lo cual es de utilidad para los diseñadores de éste tipo de redes.

CAPÍTULO VI

CONCLUSIONES, DISCUSIÓN Y RECOMENDACIONES

Conclusiones

Se desarrolló un algoritmo de enrutamiento para la reducción del consumo de energía global en una red de sensores inalámbricos basado en la Meta-heurística de Optimización de Colonia de Hormigas, el cual usa agentes (hormigas) que exploran el espacio de soluciones en la búsqueda de una ruta óptima para la transmisión de la información obtenida por los sensores.

Primeramente, se llevó a cabo una recopilación de información sobre trabajos previos y actuales en los que se proponen alternativas de enrutamiento para minimizar el consumo de energía en una red de sensores inalámbricos. A partir de allí, se propuso una solución al problema de búsqueda de rutas que minimicen el consumo de energía de la red, mediante el uso de un algoritmo de optimización basado en la metaheurística de colonia de hormigas.

La investigación consideró el consumo de energía tanto en la transmisión como en la recepción de datos, ya que por ambas vías el circuito interno del sensor gasta energía. Este aspecto es fundamental para alargar el tiempo de funcionamiento de la red y en la actualidad, debe ser incorporado en el diseño de algoritmos de enrutamiento RSI.

A diferencia de otros trabajos relacionados (Camilo et. al., 2006; Wen et. al., 2007), se estableció una heurística para el algoritmo que permite que los nodos con

mayor energía sean elegidos con mayor probabilidad por las hormigas. Adicionalmente, se determinó la cantidad de rastros de feromona a depositar por las hormigas como una función de la energía consumida, generando un mayor tráfico por las rutas con menor disipación de energía.

Por otra parte, se eligió una configuración RSI con topología de malla, a diferencia de algunos trabajos (Heinzelman et. al., 2000; Lindsey y Raghavendra, 2002; Camilo et. al., 2006), lo cual es ventajoso para casos donde todos los nodos son visibles entre sí, ya que al aplicar el algoritmo desarrollado, se pueden obtener rutas alternas con bajos niveles de energía, creando una tolerancia a fallos permitida por la redundancia de la red. De esta forma, con el itinerario del recorrido, cada sensor tiene la tarea de adquirir información del ambiente, fusionar la data y transmitirla de nodo en nodo, hasta regresar a la estación base.

Los experimentos realizados demuestran la convergencia del algoritmo de Colonia de Hormiga, ya que las mejores soluciones obtenidas por el algoritmo, a medida que transcurren las iteraciones tienden a reducir el consumo de energía a un mismo valor, lo que sugiere que el algoritmo es capaz de encontrar una ruta con bajo consumo de energía.

El aporte principal de esta investigación a las ciencias de computación es la búsqueda de soluciones óptimas a problemas reales de enrutamiento así como su posibilidad de implementación en las RSI, lo cual demuestra una vez más la efectividad de los algoritmos de Colonia de Hormigas. Finalmente, en el área de investigación de redes de sensores inalámbricos se presenta una solución en la configuración tipo malla, para la búsqueda de rutas de bajo consumo total de energía, lo cual es de utilidad para los diseñadores de éste tipo de redes.

Recomendaciones

El propósito del presente trabajo de investigación fue desarrollar un Algoritmo basado en la Meta-heurística de Colonia de Hormigas para conseguir rutas de bajo consumo de energía en redes de Sensores Inalámbricos. Por tanto, su implementación puede ser utilizada dentro de un protocolo de enrutamiento donde se tomen en cuenta nuevas variables, tales como el tiempo de vida útil de los sensores o las rutas que se hacen inaccesibles por sensores que se encuentran fuera de alcance en el radio de comunicación inalámbrica.

Como trabajo futuro el algoritmo podría ampliarse a la optimización de múltiples objetivos. La minimización de la pérdida de caminos generada al ser consumida la energía de los sensores y la maximización de la exactitud en la detección de la información ambiental adquirida por los sensores pueden ser objetivos adicionales a optimizar mientras se mantiene el consumo mínimo total de energía de la red. Así, se podrán realizar comparaciones con algoritmos de optimización, entre los que se cuentan Algoritmos Genéticos Multiobjetivos (Kalyanmoy, 2001).

Para corroborar los resultados obtenidos por las simulaciones en C++, se propone utilizar el simulador de red **ns-2** para simular la comunicación de los sensores y sus valores de energía, lo cual daría una mejor imagen del desempeño del algoritmo OCH y sustentaría la posibilidad de implementación de dicho algoritmo en redes reales de sensores inalámbricos.

GLOSARIO DE TÉRMINOS TÉCNICOS

Agente de software: Es un programa informático que habita en ordenadores y redes y que ayuda a los usuarios en tareas relacionadas con el uso del ordenador. La idea es que los agentes no son estrictamente invocados para una tarea, sino que se activan ellos mismos.

Algoritmos: Método efectivo para resolver un problema expresado como un conjunto finito de instrucciones.

Función Fitness: Es un tipo particular de función objetivo que prescribe la optimalidad de una solución (el cual es el cromosoma) en un algoritmo genético, de manera que tal cromosoma en particular pueda tener un mérito relativo a todos los demás cromosomas.

Función Objetivo: Es una expresión matemática lineal que representa el objetivo del problema.

Fenotipo: Son características observables tales como morfología, desarrollo, propiedades biomédicas, productos de comportamientos, que resultan de la expresión de los genes de un organismo como la influencia y la interacción ambiental.

Feromona: son sustancias químicas segregadas por un individuo con el fin de provocar un comportamiento determinado en otro individuo de la misma u otra especie.

Genotipo: Instrucciones heredadas o genes que se llevan en el código genético.

Neuronas: célula excitable eléctricamente que procesa y transmite información por señales químicas y eléctricas y forman parte principal del sistema nervioso.

Optimización: elección del mejor elemento de algún conjunto de alternativas disponibles.

Permafrost: (En Geología) Capa del suelo permanentemente congelada en las regiones polares.

ANEXOS

Estructura de las hormigas

```
typedef struct {
    long int  *ruta;
    char      *visitado;
    double    *energia_cons;
} est_hormiga
```

Estructura de los nodos

```
struct nodo {
    long int n;
    double x;
    double y;
    double energia;
};
```

Estructura de instancia del problema

```
struct problem{
    long int    optimum;
    long int    n; //numero de
sensores
    struct nodo *nodoptr; //sensores
    double      **distancia; //distancia entre
    nodos i y j
    double      energia_red;
    long int    **lista_nn; //lista de vecinos
mas cercanos
};
```

Procedimiento Principal

```
int main() {
    long int i;
    iteracion = 1;
    reporte = fopen("report.txt", "w+");
    reporte_comp = fopen("report_comp.txt", "w+");
    reporte_estad = fopen("report_estad.txt", "w+");
    reporte_resumen = fopen("report_resumen.txt", "w+");
    iniciar_programa(instancia, n, reporte, reporte_comp,
reporte_estad);
    semilla = (long int)
time(NULL);
```

```

instancia.lista_nn
calcula_lista_vecinos_mas_cercanos_nn(n, instancia);
feromona = generar_matriz_doble( n, n );
total = generar_matriz_doble( n, n );
inicia_intento(experimento_n,n);
    while ( iteracion < max_iter ) {
        construir_caminos(n, instancia);
        actualizar_consumo_energia();
        actualizar_estadisticas(n);
        actualizacion_feromona(n);
        control_de_busqueda_y_estadisticas(instancia, n,
        reporte_estad);
        iteracion++;
    }
    mostrar_solucion(instancia, n, reporte_estad);
    salida_intento(experimento_n, n, instancia,reporte,
    reporte_comp, reporte_estad);
}
}

free( instancia.distancia );
free( instancia.lista_nn );
free( feromona );
free( total );
free( mejor_en_intento );
free( mejor_encontrado_t );
free( tiempo_mejor_encontrado );
free( tiempo_corrida_total );
for ( i = 0 ; i < m ; i++ ) {
    free( hormiga[i].ruta );
    free( hormiga[i].visitado );
}
free( hormiga );
free( (*mejor_hormiga_hasta_ahora).ruta );
free( (*major_hormiga_hasta_ahora).visitado );
free( prob_of_seleccion );
fflush(reporte);
fflush(reporte_comp);
fflush(reporte_estad);
fclose(reporte);
fclose(reporte_comp);
fclose(reporte_estad);
return(0);
}

```

Procedimiento ConstruirCaminos

```
void construir_caminos( long int n, problem _instancia)
{
    long int k;
    long int paso;
    //marcar todos los sensores como no visitados
    for ( k = 0 ; k < m ; k++) {
        vaciar_memoria_hormigas( &hormiga[k] , n);
    }
    paso = 0;
    long int inicio = 0;
    for ( k = 0 ; k < m ; k++ )
        place_ant( &hormiga[k], paso, inicio);
    while ( paso < n-1 ) {
        paso++;
        for ( k = 0 ; k < m ; k++ ) {
            selecciona_vecino_siguiete( &hormiga[k], paso, n,
            _instancia);
            actualizacion_local_feromona( &hormiga[k], paso , n,
            _instancia);
        }
    }
    paso = n;
    for ( k = 0 ; k < m ; k++ ) {
        hormiga[k].ruta[n] = hormiga[k].ruta[0];
        actualizacion_local_feromona( &hormiga[k], paso , n ,
        _instancia);
    }
    ruta_n += m;
}
}
```

Procedimiento ActualizarConsumoEnergia

```
void actualizar_consumo_energia(){
    for (int i=0; i<m;i++){
        energia_promedio_ruta(&hormiga[i], n, instancia);
    }
};
```

Procedimiento de cálculo de energía de transmisión por paso

```
double energia_paso(est_hormiga *a, long int &paso, problem
instancia){
    long int    j, k;
    double energiapaso;
    j = (*a).ruta[paso];
    k = (*a).ruta[paso -1];
    energiapaso = Eelec*((double)bits_data*(double)paso)+
Eamp*((double)bits_data*(double)phase)*(instancia.distancia[k]
[j])*(instancia.distancia[k][j]);;
    return energiapaso;
}
```

Procedimiento de cálculo de energía de recepción por paso

```
double energia_recepcion(est_hormiga *a, long int &paso,
problema instancia){
    double energiapaso;

    stepenergy = Eelec*((double)bits_data*(double)paso);
    return energiapaso;
}
```

Procedimiento para calcular la energía promedio de la ruta

```
double energia_promedio_ruta(est_hormiga *a, long int &paso,
problem instancia) {
    long int i, j, k, h;
    double prom_energia = 0.0;
    double energia = 0.0;
    for ( i = 1 ; i <= paso ; i++ ) {
        j = (*a).ruta[i-1];
        h = (*a).ruta[i];
        instancia.nodeptr[j].energia -= energia_paso(a,
i, instancia); //Energia de Transmision
        instancia.nodeptr[h].energia -= energia_recepcion(a,
i, instancia); //Energia de Recepcion
        prom_energia +=
(instancia.nodeptr[j].energia+instancia.nodeptr[h].energia);
        ConsEnergy += energia_paso(a, i,
instancia)+energia_recepcion(a, i, instancia);
    }
    prom_energia = prom_energia/paso;
}
```

```

    (*a).energia_cons=instancia.energia_red-prom_energia;
    return prom_energia;
}

```

Procedimiento de actualización local de feromona

```

void actualizacion_local_feromona( ant_struct *a, long int
paso,long int &n, problem &instancia )
// incrementa las feromonas de los arcos en la ruta de la
hormiga k
{
    long int h, j;
    double d_tau;
    double fero,tot,step_energ;
    j = (*a).ruta[paso];
    h = (*a).ruta[paso-1];
    feromona[j][h] = (1. - eps) * feromona[j][h] +
eps*trail_0;
    feromona[h][j] = feromona[j][h];
    tot = pow(feromona[h][j], alpha) *
pow(heuristica(h,instancia),beta);
    total[h][j] = tot;
    total[j][h] = total[h][j];
}

```

Procedimiento principal de actualización de feromona

```
void actualizacion_feromona( long int n )
//deposito y actualizacion global de feromonas
{
    actualizacion_glob_fero(mejor_hormiga_hasta_ahora , n ,
    instancia);
}

```

Procedimiento de actualización global de feromona

```
// incrementa las feromonas de los arcos en las rutas de las k
hormigas
void actualizacion_glob_fero(est_hormiga *a, long int &n, problem
&instancia)
{
    long int i, j, h;
    double d_tau;
    d_tau = 1.0 / (*a).energia_cons;
    for ( i = 0 ; i < n ; i++ ) {
        j = (*a).ruta[i];
        h = (*a).ruta[i+1];
        feromona[j][h] = (1. - rho) * feromona[j][h] +
rho*d_tau;
        feromona[h][j] = feromona[j][h];
        total[h][j] = pow(feromona[h][j], alpha) *
pow(heuristica(j,instancia),beta);
        total[j][h] = total[h][j];
    }
}

```

REFERENCIAS BIBLIOGRAFICAS

- The Oficial Bluetooth Technology Web Site*. 2011. <http://www.bluetooth.com>. Ultima vez visitado en: 21/02/2011.
- Al-Karaki Jamal, Kamal Ahmed (2005). *Routing Techniques in Wireless Sensor Networks: A Survey* ICUBE, Iowa State University.
- Aydin, M.E., Fogarty T.C. (2004). *A Distributed Evolutionary Simulated Annealing. Algorithm for Combinatorial Optimization Problems*, en *Journal of Heuristics*, vol.24, no.10.
- Ballardie, A. (1997) *Core Based Trees (CBT version 2) Multicast Routing: Protocol Specification*," RFC 2189, <http://www.rfc-editor.org/rfc/rfc2189.txt> Fecha de visita: 23/08/2010.
- Bambos N. (1998). *Toward Power-Sensitive Network Architectures in Wireless Communications: Concepts, Issues, and Design Aspects*. IEEE Personal Communications.
- Bellman, R. E. (1957). *Dynamic Programming*. Princeton University Press.
- Bertsekas, D. y Gallager, R. (1992). *Data Networks*, 2da ed. Eglewood Cliffs, Prentice Hall.
- Blair, P., Polyzos, G. C., Zorzi, M.,(2001). *Plane Cover Multiple Access: A New Approach to Maximizing Cellular System Capacity*. IEEE Journal on Selected Areas in Communications, Vol. 19, No. 11.
- Camilo, T.; Carreto, C.; Silva, J.; Boavida, F. (2006). *An Energy-Efficient Ant Base Routing Algorithm for Wireless Sensor Networks*. En ANTS 2006 - Fifth International Workshop on Ant Colony Optimization and Swarm Intelligence.
- Cao G. y La Porta T., (2009). *NeTS:Small:Supporting Multi-Missions in Wireless Sensor Networks*, Programa NSF: RES IN NETWORKING TECH & SYS. National Science Fundation.
- Coltun, R.; D. Ferguson, J Moy y A. Lindem (2008). "OSPF for IPv6". Internet Engineering Task Force. <http://www.ietf.org/rfc/rfc5340.txt>. Fecha de visita: 23/08/2010
- Cook, W. (2010). <http://www.tsp.gatech.edu/index.html>. Fecha de visita: 21/06/2010.
- Cormen T., Leiserson C., Rivest R. y Stein C. (2001). *Introduction to Algorithms*. Massachusetts Institute of Technology. Mc. Graw Hill. 2da Edición.

- De Smith M J, Goodchild M F, Longley P A (2007) *Geospatial analysis: A comprehensive guide to principles, techniques and software tools*. 2da edition. Troubador, UK.
- Deb Kalyanmoy (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley and Sons.
- Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". *Numerische Mathematik* 1: 269–271.
- Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan.
- Dorigo, M., & Gambardella, L. M. (1996). *A study of some properties of Ant-Q*. In H. Voigt, W. Ebeling, I. Rechenberg, & H. Schwefel (Eds.), *Proceedings of PPSN-IV, Fourth International Conference on Parallel Problem Solving from Nature*, vol. 1141 of *Lecture Notes in Computer Science* (pp. 656–665). Berlin, Springer-Verlag
- Dorigo, M., Stützle, T. (2004). *Ant Colony Optimization*. MIT Press. ISBN 0262042193.
- Dousse O., Tavoularis C., y Thiran P, 2006. *Delay of Intrusion Detection in Wireless Sensor Networks,*” In *Proc. of the 7th of ACM International Symposium on Mobile Ad Hoc Networking and Computing*. pp.155–165. MobiHoc’06. ACM 1-59593-368-9/06/0005. Florencia, Italia.
- Eidgenössische Technische Hochschule Zürich (2006). Permasense. <http://www.permasense.ch>. Fecha de visita: 26/06/2010.
- Banner Engineering (2009), Application Notes http://www.bannerengineering.com/en-US/wireless/surecross_web_appnotes. Fecha de visita: 02/07/2010.
- Fall K. y Varadhan K. *The ns Manual* (Manual de ns), <http://www.isi.edu/nsnam/ns/ns-documentation.html>, ultima vez visitado: Febrero 2010.
- Ford, L. R. Jr. Y Fulkerson, D.R. (1962). *Flows in Networks*. Princeton University Press.
- Garey M. R., Johnson D. S. (1979). *Computers and Intractability: A guide to the theory of NP-completeness*. Freeman&Co. ISBN 0716710447.
- Glover, F., & Laguna, M. (1997). *Tabu Search*. Boston, Kluwer Academic Publishers.

- Goldberg D. (1989). *Genetic Algorithm is Search, Optimization & Machine Learning*. Addison Wesley.
- Hart J. K., Martinez K. (2006). *Environmental Sensor Networks: A revolution in the earth system science?*. Earth-Science Reviews, pp. 177-191.
- Hasler A., Talzi I., Beutel J., Tschudin C. y Gruber S. (2008). *Wireless sensor networks in permafrost research - concept, requirements, implementation and challenges*. In Proc. 9th Int'l Conf. on Permafrost (NICOP 2008), vol. 1, pp. 669–674.
- Heinzelman W., Chandrakasan A., y Balakrishnan H. (2000) *Energy-Efficient Communication Protocol for Wireless Microsensor Networks*. In Proceedings of the Hawaii Conference on System Sciences.
- Holland J. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press Edition.
- Kennedy, J.; Eberhart, R.C. (2001). *Swarm Intelligence*. Morgan Kaufmann. ISBN 1-55860-595-9.
- Kirkpatrick, S., Gelatt, C. D., Jr., & Vecchi, M. P. (1983). *Optimization by simulated annealing*. Science, 220, 671–680.
- Kröse, B., Smagt P (1996). *An introduction to Neural Networks*. University of Amsterdam.
- Lewis, F.L., (2004). “*Wireless Sensor Networks,*” *Smart Environments: Technologies, Protocols, and Applications*, ed. D.J. Cook and S.K. Das. John Wiley. New York.
- Lindsey S., Raghavendra C. S. (2001). *PEGASIS: Power Efficient Gathering in Sensor Information Systems*. ICC 2001.
- Lourenc,o, H., y Serra, D. (2002). *Adaptive search heuristics for the generalized assignment problem*. Mathware and Soft Computing, 9(2–3), 209–234.
- Martinez K. y Hart J.K. (2010). *Glaciers Monitoring: Deploying Custom Hardware in Harsh Environments*. In: *Wireless Sensor Networks - Deployments and Design Frameworks*, Springer. ISBN 978-1-4419-5833-4
- McQuillan, John M., Richer. Isaac y C. Rosen, Eric (1979). *An overview of the new routing algorithm for the ARPANET*. SIGCOMM '79: Proceedings of the sixth symposium on Data communications. ACM.

- MEMSIC Corporation (2010). *The Memsic Solution*, <http://www.memsic.com/solutions/environmental.html>.
- Nakamura E., Loureiro A., Frery A. (2007). Information fusion for wireless sensor networks: Methods, models, and classifications, *ACM Computing Surveys*, Volume 39, Issue 3, Article 9.
- Okdem, S.; Karaboga, D. (2009). *Routing in Wireless Sensor Networks Using An Ant Colony (OCH) Router Chip*. *Sensors* 9, no. 2: 909-921.
- Rajagopalan R., Chilukuri M., Pramod V. and Kishan M. (2005). Multi-objective Mobile Agent Routing in Wireless Sensor Networks. In *Proceedings of IEEE Congress on Evolutionary Computation*.
- Ramanathan, R. y Hain, R. (2000). *Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment*. En *Proceedings Infocom 2000*.
- Rappaport T. S. (1996). *Wireless Communications Principles and Practice*. 1st. Edition, Prentice Hall. New Jersey.
- Steele R. (1992). *Mobile Radio Communications*, Pentech Press, John Wiley.
- Suzuki M., Saruwatari S., Kurata N. y Morikawa H. 2007. *A high-density earthquake monitoring system using wireless sensor networks*. *SenSys '07 Proceedings of the 5th international conference on Embedded networked sensor systems ACM*. ISBN: 978-1-59593-763-6, New York.
- Universidad Centroccidental Lisandro Alvarado (UCLA) (2002). *Manual para la Presentación de Trabajo de Grado*. Barquisimeto. Material Mimeografiado.
- Vázquez Iñaki (2007). *Experiencias en la aplicación de Wireless Sensor Networks*. MoreLab - Mobility Research Lab. En *IV Jornadas sobre tecnologías Inalámbricas*. Zamudio.
- Walpole R. y Myers R.(1992). *Probabilidad y Estadística*. Cuarta Edición. McGraw-Hill. México.
- Y. Wen, Y. Chen y D. Qian. (2007). *An Ant-Based Approach to Power-Efficient Algorithm for Wireless Sensor Networks*. *Proceedings of the World Congress on Engineering*, Vol. II, pp. 1546-1550.