

**UNIVERSIDAD CENTROCCIDENTAL
“LISANDRO ALVARADO”**

**ALGORITMO GENÉTICO MULTI OBJETIVO “SPEA” PARA EL
ENRUTAMIENTO DE AGENTES EN REDES DE SENSORES
INALÁMBRICOS HETEROGÉNEOS**

SANDRA LIMA CASTRO

Barquisimeto, 2010

**UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”
DECANATO DE CIENCIAS Y TECNOLOGIA**

**ALGORITMO GENÉTICO MULTI OBJETIVO “SPEA” PARA EL
ENRUTAMIENTO DE AGENTES EN REDES DE SENSORES
INALÁMBRICOS HETEROGÉNEOS**

Trabajo presentado para optar a la categoría de Asistente

SANDRA LIMA CASTRO

Barquisimeto, 2010

**ALGORITMO GENÉTICO MULTI OBJETIVO “SPEA” PARA EL
ENRUTAMIENTO DE AGENTES EN REDES DE SENSORES
INALÁMBRICOS HETEROGÉNEOS**

Por: SANDRA LIMA.

Trabajo de Asenso aprobado

Nombre:
CI

Nombre:
CI

Nombre:
CI

Barquisimeto, ____ de _____ de 2010
(Fecha de defensa)

INDICE GENERAL

	PAG.
INDICE DE FIGURAS	v
INDICE DE ALGORITMOS	vi
RESUMEN	vii
INTRODUCCION	1
CAPITULO	
I ELPROBLEMA	4
Planteamiento del Problema	4
Objetivos	9
Objetivo General	9
Objetivos Específicos	9
Justificación e Importancia	9
Alcance y Limitaciones	10
II MARCO TEORICO	12
Antecedentes de la Investigación	12
Bases Teóricas	16
III MARCO METODOLOGICO	40
Tipo de Investigación	40
Estudio de Factibilidad	41
Técnica	41
Económica	41
Procedimiento de la Investigación	41
IV DISEÑO EN INGENIERÍA DE LA PROPUESTA	43
Formulación Matemática de los Objetivos a Optimizar	43
Simulación de la Red	47
Representación de los Cromosomas	49
Mecanismo para evaluar las Funciones Objetivos a optimizar	50
Descripción del Algoritmo	50

V	EXPERIMENTOS Y RESULTADOS	56
	Conclusiones	62
	Recomendaciones y Trabajos Futuros	64
	GLOSARIO DE TÉRMINOS TÉCNICOS	66
	REFERENCIAS BIBLIOGRAFICAS	70

INDICE DE FIGURAS

FIGURA		PAG.
1	<i>Arquitectura de Redes de Sensores Inalámbricos.</i>	6
2	<i>Funcionamiento general de Un Algoritmo Genético.</i>	31
3	<i>Resultados del Frente de Pareto.</i>	56
4	<i>Minimización del Consumo total de Energía. 100 Experimentos.</i>	57
5	<i>Minimización de la Pérdida Total de Caminos. 100 Experimentos.</i>	57
6	<i>Maximización Exactitud en la Detección. 100 Experimentos.</i>	58
7	<i>Minimización del Consumo total de Energía. 800 Generaciones.</i>	58
8	<i>Minimización de la Pérdida Total de Caminos. 800 Generaciones.</i>	59
9	<i>Maximización Exactitud en la Detección. 800 Generaciones.</i>	59
10	<i>Promedio Funciones Objetivos y Desviación Estándar.</i>	59
11	<i>Relación Cartesiana entre la Pérdida Total de Caminos y Exactitud en la Detección</i>	60
12	<i>Relación Cartesiana entre la Pérdida Total de Caminos y el Consumo de Energía</i>	61
13	<i>Comparación Metrica “Spacing” entre EMOCA, NSGA-II y el Algoritmo Propuesto (AP). 800 Generaciones.</i>	61

INDICE DE ALGORITMOS

ALGORITMO	PAG.
1 <i>Pseudocódigo del Algoritmo Dijkstra.</i>	18
2 <i>Procedimiento de Optimización Evolutiva.</i>	28
3 <i>Pseudocódigo del Algoritmo Genético para Enrutamiento de Sensores Inalámbricos.</i>	51
4 <i>Pseudocódigo del procedimiento Marcar individuos no dominados en P.</i>	52

UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”
DECANATO DE CIENCIAS Y TECNOLOGIA

**ALGORITMO GENÉTICO MULTI OBJETIVO “SPEA” PARA EL
ENRUTAMIENTO DE AGENTES EN REDES DE SENSORES
INALÁMBRICOS HETEROGÉNEOS**

POR: Sandra Lima Castro

RESUMEN

La fusión de sensores es una técnica que combina la información provista por diferentes sensores en una red, para obtener una mejor imagen del entorno. El resultado de fusionar data conduce a mayor integridad, exactitud y confianza. Para cumplir esta misión, Un agente viaja a través de los sensores, integra los datos de cada uno y lleva el resultado final a la estación central para ser procesada. Sin embargo, por ser sensores con diferentes capacidades y características, debe existir un equilibrio entre las funciones objetivo: Consumo de Energía, Pérdida de Caminos y Exactitud en la Detección. Los algoritmos evolutivos multiobjetivo, han demostrado ser una excelente alternativa cuando se trata de optimizar objetivos en conflicto, por su robustez en la exploración del espacio de búsqueda y su paralelismo para obtener soluciones en un tiempo razonable. La presente investigación tiene como propósito desarrollar un algoritmo genético multiobjetivo basado en SPEA (Strength Pareto Evolutionary Algorithm) para la determinación de un conjunto de rutas óptimas que un agente móvil pueda tomar para fusionar datos en redes de sensores inalámbricos heterogéneos, el cual contribuye a alargar la vida de la red y evitar su congestión.

Palabras Claves: Redes de Sensores Inalámbricos, Algoritmo Genético Multiobjetivo, SPEA, Fusión de Sensores, Agente, Algoritmos Evolutivos.

INTRODUCCIÓN

Las redes de sensores constituyen la clave para recopilar información en entornos inteligentes. Un entorno inteligente se alimenta de datos del mundo real, los cuales provienen de múltiples sensores de diferentes características distribuidos en el espacio. Por ello, las redes de sensores inalámbricos están tomando cada vez más importancia en el área de medicina, seguridad, control de tráfico, entre otros, donde las distintas formas de detección son complementarias e independientes, y algunos sensores pueden capturar información no disponible a otros para obtener una mejor imagen del ambiente. En conducción autónoma de vehículos, el ruido de radiofrecuencia que pudiera afectar a los sensores por radar no influye en los sensores de visión. A su vez, los sensores de visión pueden localizar señales de tránsito en la carretera, a diferencia de los sensores por radar.

Si se logra combinar y procesar toda la información capturada, se obtendría mayor precisión del ambiente inspeccionado. Bajo este contexto, fusionar data en una red de sensores, conduce a mayor integridad, exactitud y confianza. Sin embargo, por su naturaleza inalámbrica y heterogénea, los nodos de la red poseen radios de cobertura limitada, carecen de la capacidad de cómputo necesaria y no cuentan con mucha carga de batería para enviar la data directamente a la estación central, donde será procesada.

El desarrollo de agentes provee una infraestructura mejorada frente a la arquitectura cliente-servidor: La data permanece en el sitio local, mientras el agente se mueve a los nodos ahorrando ancho de banda y proporcionando una forma efectiva de superar el congestionamiento de la red. Por tanto, se hace evidente la necesidad de diseñar un algoritmo de enrutamiento que pueda ser adaptable a las características de la red y al mismo tiempo prolongar su vida útil tanto como sea posible. El tiempo de vida de una red de sensores inalámbricos puede medirse basado en parámetros

genéricos, tales como el tiempo que transcurre hasta que la mitad de los sensores pierden su capacidad de transmisión, retardo mínimo o exactitud en la detección.

Aunque la mayoría de los problemas de enrutamiento involucran este tipo de situaciones, donde existen varias metas en conflicto, las propuestas computacionales presentadas hasta la fecha se limitan a convertir el problema de múltiples objetivos en uno en el que existe un solo objetivo, de acuerdo al modelo matemático empleado. Se puede modelar por ejemplo, priorizando uno de los objetivos y utilizando los restantes objetivos como restricciones, o generando un objetivo compuesto, otorgando pesos a cada uno de ellos, a fin de optimizar la suma ponderada de los mismos. De todos modos, ninguna de estas reducciones refleja fielmente al problema y por tanto, tampoco otorga soluciones completamente satisfactorias.

En el presente trabajo se propone un algoritmo evolutivo de enrutamiento basado en SPEA (Strength Pareto Evolutionary Algorithm), el cual tomando en cuenta todos los objetivos simultáneamente, pueda encontrar un conjunto de caminos óptimos para la migración del agente móvil, de manera que ahorre la máxima energía posible y alargue la vida de la red.

Este proyecto de investigación está estructurado en capítulos. En el primero (El Problema), se exponen los diferentes elementos que conforman el entorno de la situación planteada con respecto a los avances tecnológicos en materia de fusión de sensores y el papel de los agentes en dicho proceso. Igualmente, se describen los objetivos del mismo, la justificación e importancia, alcances y limitaciones.

El Capítulo 2 (Marco Teórico) se inicia con los antecedentes y las bases teóricas que apoyan el conocimiento del tema en estudio; se describen investigaciones realizadas para fusión de datos multisensores, conceptos asociados a los algoritmos de enrutamiento, problemas de optimización multiobjetivo y técnicas de inteligencia artificial tales como Algoritmos Evolutivos en optimización multiobjetivo, entre los que se destaca SPEA.

En el Capítulo 3 (Marco Metodológico) se describe la naturaleza del estudio, el procedimiento de la investigación y su estudio de factibilidad. El Capítulo 4 (Diseño e Ingeniería de la Propuesta) detalla los aspectos asociados al algoritmo propuesto, especificando las metas del problema de enrutamiento y actividades realizadas. Finalmente en el capítulo 5 (Experimentos y Resultados) se presentan las conclusiones y recomendaciones finales de la investigación.

CAPITULO I

EL PROBLEMA

Planteamiento del Problema

A nivel mundial se continúan estudiando tecnologías para la construcción de máquinas “inteligentes” que ejecuten determinadas tareas de forma autónoma. El tren AVE (Alta Velocidad Española) es una muestra de ello, y aunque posee dos formas de conducción, se considera más arriesgada la conducción humana. De hecho Gimeno (2010) reporta que en caso de conflicto entre las instrucciones dadas por el operario y las generadas por el ordenador, prevalece lo ordenado por la máquina.

El proyecto de la NASA (2004) X-45 UCAV (Unmanned Combat Air Vehicle, o avión de combate sin piloto), también constituye un éxito en el ámbito de la aviación militar. Se trata de un aparato que se conecta con una estación de control y tiene un peso, tamaño y precio menor a los aviones tripulados. El rodaje, despegue y aterrizaje son totalmente autónomos, pero hay la posibilidad de que un piloto-operador controle estas maniobras. Las misiones para las que está diseñado este prototipo son de alto riesgo, supresión de defensas aéreas o aquellas consideradas de alta prioridad en el ataque. Esta autonomía se ha logrado instalando redes de sensores que puedan inspeccionar el ambiente en tiempo real, y una vez procesada la información, se ejecuten las acciones ordenadas por la estación central.

La fusión de sensores es una técnica que combina la información provista por diferentes sensores en una red, para obtener una mejor imagen del entorno (Nirupama y Sanjay, 2005). Al contar con múltiples sensores, aparte de proporcionar diferentes coberturas, pueden detectar por ejemplo, el mismo objeto pero con distintos niveles

de exactitud en los parámetros que lo describen. Así, el rango de precisión de un escáner láser es mucho mayor que la gama de los sensores derivados de la visión (Nirupama and Sanjay, 2005). Por tal motivo, dicha técnica puede ser utilizada como tecnología base para una gran variedad de aplicaciones.

El desarrollo de Agentes Móviles para Redes de Sensores Distribuidos (MADSNs, por sus siglas en inglés) forman una infraestructura mejorada para la fusión de datos multisensores (Wu et al., 2004). Comparado con la arquitectura cliente-servidor, los agentes móviles adoptan un nuevo modelo computacional: La data permanece en el sitio local, mientras el agente se mueve a los nodos ahorrando ancho de banda y proporcionando una forma efectiva de superar los problemas de tráfico de paquetes en la red.

De esta manera, un agente selectivamente visita los sensores e incrementalmente fusiona data. Una vez visitados todos los nodos de la red, entrega la data recolectada a la estación central donde posteriormente será procesada. Los resultados publicados en Qi et al. (2001), han demostrado que la implementación de un agente móvil guarda casi el 90% de la data transferida evitando el congestionamiento de la red. El destino del agente es fijo (la estación central) pero puede ser disparado desde cualquier nodo fuente, previamente calculado por la estación central.

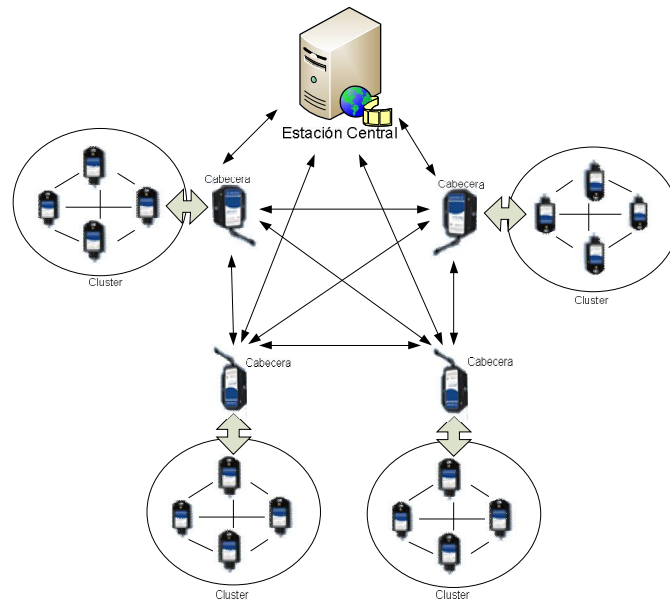


Figura 1: *Arquitectura de Redes de Sensores Inalámbricos.*
(Rajagopalan et al., 2005)

En la Figura 1, se puede observar una Red de Sensores Inalámbricos (WSN, Wireless Sensor Network), la cual está formada por clústeres, y cada uno compuesto por sensores. Estos se encuentran aleatoriamente distribuidos y recolectan la misma información (por ejemplo, medidas acústicas, sísmicas e infrarrojas), ya que algunos sensores pueden ver información "invisible" a otros para obtener mayor integridad, exactitud y confianza en la data recolectada. Cada sensor debe enviar dicha data al nodo "cabecera de cluster", el cual es un sensor con características especiales, tales como mayor capacidad de procesamiento, mayor vida de batería en comparación al resto, y de allí pasar a la estación central, donde se procesará la data fusionada.

Para recorrer todos los nodos, el agente debe conocer el itinerario con la ruta de migración, la cual puede ser fija o determinada dinámicamente de acuerdo al status actual de la red. Seleccionar una ruta entre dos nodos de una red es uno de los aspectos más importantes, sobre todo si se trata de sensores inalámbricos. Esta tarea es llevada a cabo por los algoritmos de enrutamiento, los cuales se encargan de elegir la ruta más adecuada.

En este escenario, los nodos cabeceras y la estación central tienen predeterminado el conocimiento necesario para computar la ruta, tales como coordenadas geográficas (utilizando GPS) y parámetros de transmisión/recepción de los nodos sensores. La estación central calcula el *camino inter-clúster* que consiste en una secuencia no cíclica de cabeceras de clúster, es decir un camino donde se consideran todos los nodos sin repeticiones. Para cada cabecera de clúster en el camino inter-clúster, también se calcula un *camino intra-clúster* que consta de una secuencia no cíclica de sensores dentro de dicho clúster. El agente es enviado desde la estación central con el itinerario y visita el primer nodo cabecera en el camino, seguido por la secuencia de nodos sensores dentro de ese clúster, recolecta la data, y retorna al nodo cabecera. Luego sucesivamente visita los restantes nodos cabeceras y nodos sensores en la secuencia de enrutamiento y regresa a la estación central.

Para construir ese itinerario óptimo, considerando la presencia de sensores heterogéneos, donde cada nodo posee diferentes niveles de carga de batería, capacidades de adquisición y procesamiento de datos, y distancia a la fuente emisora de la señal que registra, se hace necesario minimizar el consumo total de energía para aumentar el tiempo de funcionamiento de la red, minimizar la pérdida total del camino para evitar la atenuación de la señal inalámbrica y también asegurar que la data recolectada por el agente sea confiable.

En consecuencia, en la determinación de la ruta, debe existir un equilibrio entre estas tres funciones objetivos: Consumo de Energía, Pérdida del Camino y Exactitud en la Detección (Rajagopalan et al., 2005). Por ejemplo, si el número de sensores se incrementa, la calidad de la data fusionada mejora pero el consumo de energía y la pérdida de caminos se incrementan. Adicionalmente, mientras mayor sea el tamaño de la data a transmitir, el consumo de energía aumenta, lo cual puede provocar un mayor número de caminos perdidos entre nodos que a su vez degradaría la señal de comunicación con el resto de la red.

Ante lo planteado surgen algunas interrogantes a ser resueltas durante el desarrollo del presente trabajo:

1. Tomando en cuenta la naturaleza antagónica de los objetivos a optimizar, ¿Es posible encontrar alguna ruta que los satisfaga simultáneamente bajo un nivel de confianza determinado?
2. ¿Cómo sería el mecanismo para evaluar las funciones objetivos en el camino inter-cluster e intra-cluster?
3. ¿Cómo construir un escenario de simulación de la red para evaluar la calidad de las diferentes rutas?

Los algoritmos de optimización multiobjetivo, representan una excelente alternativa cuando se intenta combinar varios objetivos en conflicto (Wu et al., 2004). En especial, los algoritmos evolutivos son capaces de encontrar en una sola corrida múltiples soluciones y no requieren conocimiento previo del problema a resolver, a diferencia de los métodos clásicos en los que no se obtiene un efecto sinérgico (Deb, 1999).

El presente trabajo describe un algoritmo genético multiobjetivo (Zitzler and Thiele, 1998) basado en SPEA (Strength Pareto Evolutionary Algorithm), diseñado y parametrizado para evaluar simultáneamente los objetivos antes planteados, dando como resultado un conjunto de rutas óptimas que el agente pueda tomar. Se hace necesario simular un ambiente de configuración de red para evaluar el rendimiento del algoritmo en términos de su convergencia al Frente Pareto Óptimo. Adicionalmente se analizará el espaciamiento de las soluciones no dominadas obtenidas utilizando la métrica “Spacing” propuesta por Deb (2001) y se comparará con los resultados obtenidos por Ragajopalan et al. (2005).

Objetivos

Objetivo General

Desarrollar un Algoritmo Genético Multiobjetivo basado en SPEA para la determinación de un conjunto de rutas óptimas que un agente pueda tomar para fusionar datos en redes de sensores inalámbricos heterogéneos.

Objetivos Específicos

1. Diseñar un Ambiente de simulación para la configuración de la red de sensores inalámbricos.
2. Diseñar un mecanismo de evaluación de las funciones objetivos a optimizar en el camino inter-cluster e intra-cluster.
3. Diseñar un algoritmo genético multiobjetivo SPEA con sus parámetros correspondientes, que considere las funciones objetivos planteadas.
4. Diseñar un conjunto de experimentos que permitan evaluar el rendimiento del algoritmo propuesto.

Justificación e Importancia

Las WSN están tomando auge por sus diversas aplicaciones en el campo industrial, domótica e inmótica, salud, transporte, entre otros. Vázquez (2007) muestra algunos ejemplos:

- Obtención de medidas de consumo de agua, electricidad, para su optimización.
- Creación de entornos que se adaptan dinámicamente a las actividades de los usuarios.
- Por motivos de seguridad, prevención de riesgos laborales, o eficiencia en el uso de recursos.

- Recolección de información sobre el uso/comportamiento de los recursos para optimizar su costo y ciclo de vida.
- Supervisión de ancianos, personas con necesidades especiales.
- Monitoreo de habitats / ecología.
- Monitoreo de fuentes de contaminación, agentes químicos o biológicos.
- Monitoreo de tráfico, localización, posicionamiento.

En estas aplicaciones, los diversos modos de detección son generalmente complementarios e independientes, ya que algunos sensores pueden capturar información no disponible a otros y obtener una mejor imagen del ambiente. En conducción autónoma de vehículos, el ruido de radio-frecuencia que pudiera afectar a los sensores por radar no influye en los sensores por visión. Por otro lado, los sensores de visión pueden localizar señales en la carretera, a diferencia de los sensores por radar. En consecuencia, el resultado de fusionar data conduce a mayor integridad, exactitud y confianza.

Estas nuevas tecnologías han superado el clásico modelo basado en datagramas, en el cual cada paquete de datos es enviado en forma independiente a su destino. Ahora, el objetivo es garantizar un ancho de banda mínimo y evitar el congestionamiento de la red. Por tanto, encaminar un agente móvil a través de una red de sensores inalámbricos determinando un itinerario de migración óptimo, proporcionaría una manera efectiva de superar estas dificultades.

Alcance y Limitaciones

En esta investigación se simulará una arquitectura de red de sensores inalámbricos de diferentes tamaños y patrones de distribución. Para la construcción del algoritmo genético, la población inicial contará con individuos cuya configuración de ruta sea aleatoria y acíclica. Para evaluar el valor de adaptación o “fitness”, sólo se

tomarán en cuenta los objetivos Consumo de Energía, Pérdida de Caminos y Exactitud en la Detección. No se considera la precisión en la convergencia al conjunto de rutas óptimas, sino un número de generaciones previamente determinado para examinarla.

CAPITULO II

MARCO TEORICO

Antecedentes de la Investigación

Sistemas de Fusión de Sensores

Proyecto X-45 UCAV (Unmanned Combat Air Vehicle)

Este proyecto desarrollado por la NASA a partir del año 2000, tiene como meta demostrar que un avión de gran autonomía puede suprimir ataques aéreos y a la vez servir de espía. El avión X-45 (NASA, 2004) está equipado con un conjunto de sensores, entre los que se incluyen, vectores de escáneres electrónicos, radar de apertura sintética y un sistema electrónico de soporte desarrollado por Raytheon. El conjunto de sensores instalados permiten la detección, identificación y localización de blancos fijos y móviles en tiempo casi real. La situación del campo de batalla y los datos del blanco, son enviados a través de enlaces de comunicación satelital a la estación terrestre de control. Dicha estación está equipada con herramientas de inteligencia artificial para asistir al piloto-operador en la evaluación de la situación en el campo de batalla y su decisión de autorizar la liberación de armas.

El rodaje, despegue y aterrizaje son totalmente autónomos, sin embargo, el piloto-operador tiene posibilidad de controlar estas maniobras. La estación terrestre del UCAV ha sido diseñada por la NASA. BAE Systems Controls es la compañía encargada del sistema automatizado de gestión de la aeronave. El satélite de comunicaciones pertenece a Milstar.

Proyecto CARSENSE (Detection of Car Environment at Low Speed Driving)

CARSENSE fue culminado en el año 2002 (Nirupama and Sanjay, 2005), como un proyecto europeo que se centró en la detección de objetos de baja velocidad en el entorno automotor urbano utilizando la técnica de fusión de sensores. El proyecto CARSENSE se consideró un paso importante en el camino a los sistemas de percepción de alto rendimiento (ADAS).

Como punto de referencia para las autopistas urbanas, los investigadores utilizaron las grandes carreteras paralelas a la periferia. El tráfico en esos caminos se caracteriza por:

- Paralización del tráfico con paradas o paso lento de vehículos.
- Gran número de motocicletas cruzando y zigzagueando a través del tráfico.
- Mezcla de tráfico motorizado con bicicletas y peatones.
- Configuración de carreteras con curvas, puentes, túneles cortos, y trabajos en la zona.

Para ello, definieron un conjunto de escenarios que representan estas situaciones en un sistema de prueba:

- Avanzar en tráfico denso;
- Adelantar vehículos y frenar en una multitud de vehículos en un carril;
- Maniobras de cambio de carril con dos o más objetos involucrados;
- Mover el vehículo ante un obstáculo fijo;
- Mover el vehículo ante múltiples obstáculos fijos;
- Variedad de obstáculos (tamaño, forma, color de los objetos diferentes);
- Obstáculos transitorios que pasan de estacionario a en movimiento;
- Maniobrar en presencia de tráfico en dirección contraria.

El proyecto se centró en los algoritmos de fusión de datos y su rendimiento en la detección de obstáculos dentro de la carretera. Los sensores utilizados pertenecen a diferentes familias, entre los que se destacan aquellos basados en radar, láser, procesamiento de imágenes y video. Algunas de las innovaciones realizadas dentro del proyecto incluyen la exploración del canal paralelo y ángulos, el reconocimiento de geometría de la vía (basado en fronteras y otros objetos), y la clasificación de objetos típicos (señales de tránsito, alcabalas, entre otros).

Los sensores CARSENSE entregan la información sobre la geometría de la vía y los objetos relevantes detectados en las cercanías del vehículo de prueba, bajo la forma de una lista de objetos. Cada objeto se caracteriza por un conjunto de atributos (como la posición y velocidad), y la calidad de los mismos varía dependiendo del sensor y la operación de fusión que se realice. El algoritmo crea un mapa de localización de objetos en la parte delantera del vehículo en tiempo real, e incluye una velocidad relativa y un nivel de confianza en la precisión de la detección. Los resultados mostraron que la tasa de detección de obstáculos obtenida por fusión era significativamente mejor que si los sensores actuarán por sí solos.

Fusión de Datos Multisensores

Qi et al. (2001), presenta una infraestructura mejorada para la fusión de datos multisensores gracias a la utilización de un Agente de Software Móvil en Redes de Sensores Distribuidos (MADSN, por sus siglas en inglés). La exactitud en la integración de los datos se alcanza progresivamente durante la migración del agente. Para ello describe un Algoritmo de Integración Multirresolución (MRI), el cual consiste básicamente en la construcción de una función llamada “función de superposición” que será la salida de los sensores en un clúster. Resolver esta función iterativamente en varias escalas de resolución logra aislar la región de los sensores libres de fallas. Este algoritmo supone una ruta de migración estática para el agente sin considerar que la elección de la ruta podría afectar la efectividad de la fusión.

Específicamente, si los sensores poseen la misma naturaleza, se hace necesario tomar una decisión representativa de la región completa. Clouqueur et al. (2001), expone dos métodos de fusión de datos:

- Fusión por Decisión: Cada sensor puede tomar su decisión independiente sobre sus propios valores medidos e intercambiar sus decisiones con los otros para llegar a un consenso por fusión.
- Fusión por Valor: Los sensores de la red intercambian sus valores de energía local y los fusionan al encontrar el promedio. La decisión de detección final se hace comparando el valor final con un umbral.

De igual modo, los sensores con fallas pueden proveer valores incorrectos que afecten las decisiones sobre los otros sensores en la red. Por ello, es importante considerar otros parámetros tales como consumo de energía, número de mensajes intercambiados, o el ancho de banda requerido para llegar a un consenso efectivo al momento de fusionar.

Por su parte Rajagopalan et al. (2005), plantea la tarea de conducir agentes en una red jerárquica de sensores inalámbricos (figura 1), utilizando un algoritmo evolutivo de optimización multiobjetivo llamado EMOCA (Evolutionary Multiobjective Crowding Algorithm for Path Computations), el cual debe: (a) minimizar el consumo de energía, (b) minimizar la pérdida de la ruta y (c) maximizar el total de energía en la señal detectada. Este algoritmo introduce, además del criterio de no-dominación, un criterio de diversidad, donde el valor de adaptación de cada individuo es igual a la suma de su valor de no-dominación y su valor de diversidad. De esta manera, da oportunidad a los individuos dominados de formar parte del pool de apareo si contribuyen a la diversidad de la población. Finalmente, hace la comparación del EMOCA con NSGAI, demostrando que EMOCA es mucho más exitoso en este tipo de problemas. Sin embargo, no efectúa ninguna poda al conjunto de rutas óptimas que el agente puede tomar.

Zitzler y Thiele (1998), proponen el algoritmo genético multiobjetivo SPEA, donde el valor de adaptación de los individuos se asigna de acuerdo a la relación de dominancia de Pareto, lo cual influye en la diversidad poblacional. Adicionalmente, incorpora un método de agrupación (clustering) para reducir el conjunto de Pareto (soluciones óptimas) sin destruir sus características y de esta manera ayudar a tomar la decisión final en un tiempo razonable. Muchas investigaciones sugieren que el SPEA es muy efectivo en el muestreo del frente de Pareto Óptimo y la distribución de las soluciones generadas sobre la superficie solución. Por sus características de robustez y adaptabilidad, SPEA ha demostrado su buen funcionamiento en la resolución de problemas de ingeniería como el diseño topológico de redes de computadoras (Barán and Benitez, 2003), la programación de bombeo de agua en sistemas de suministro de agua (Sotelo et al., 2002) y la compensación reactiva de potencia (Barán et al., 2003). Por ello, en el presente trabajo se propone un algoritmo basado en SPEA para resolver el problema de enrutamiento en redes de sensores inalámbricos.

Bases Teóricas

Algoritmos de Enrutamiento

Enrutamiento o Encaminamiento consiste en determinar entre todos los posibles caminos, cuál es el mejor para una transmisión de datos desde un nodo fuente a un nodo destino (Forouzan, 2001). El mejor camino a elegir es el más corto es términos de interconexión de redes. En algunos casos, el más corto significa la ruta que requiere el menor número de retransmisiones o saltos. En otros casos, el más corto significa el más rápido, más económico, más fiable, más seguro. Frecuentemente el más corto hace una combinación de todos los anteriores.

En particular, para lograr la fusión de sensores y en consecuencia obtener mayor robustez y rendimiento, se debe seleccionar una ruta efectiva entre nodos de manera que el agente integre data en el menor tiempo posible y con el menor uso de recursos de la red. Esta tarea es llevada a cabo por los algoritmos de enrutamiento, los cuales se encargan de seleccionar la ruta más adecuada.

El enfoque mas común utilizado en redes de computadoras para enviar paquetes es el enrutamiento a través del camino más corto, considerando la métrica de retardo o número de saltos, variantes del algoritmo de Dijkstra (1959).

Algoritmo de Dijkstra

El algoritmo de Dijkstra (Cormen et al., 2001) resuelve el problema del camino mas corto en un grafo dirigido ponderado $G = (V, E)$. Se asume, sin perdida de generalidad, que la función de peso $w(u, v) \geq 0$ para cada arista $(u, v) \in E$.

Este algoritmo mantiene un conjunto S de vértices cuyo resultado final corresponde al camino de menor peso desde el nodo fuente s . Selecciona repetidamente un vértice $u \in V - S$ que contiene el camino de menor costo estimado $d[u]$, añade u a S y relaja todas las aristas v que salen de u . Para su implementación, se usa una cola de prioridad Q , cuyas claves son sus valores d .

La Técnica de Relajación se utiliza en grafos dirigidos donde para cada vértice $u \in V$, se mantiene un atributo $d[u]$, como frontera superior del peso de la ruta más corta desde la fuente s a u . A Este atributo se le conoce como el estimado de ruta más corta. La relajación de una arista (u, v) consiste en probar si se puede mejorar la ruta más corta a v encontrada a través de u , y si es posible, actualizar $d[v]$ y $\Pi[v]$. El proceso puede decrementar el valor de estimación de la ruta mas corta $d[v]$ y actualizar su predecesor $\Pi[v]$.

```

Inicio
  Inicializar nodo fuente  $(G,s)$ ;
   $S = \phi$ ;
   $Q = V[G]$ ;
  Mientras  $Q \neq \phi$  {
     $u = \text{EXTRAER-MIN}(Q)$ ;
     $S = S \cup \{u\}$ ;
    Para cada vértice  $v \in \text{Adj}[u]$  {
       $\text{RELAJAR}(u,v,w)$ ;
    }
  }
Fin

```

Algoritmo 1: Pseudocódigo del Algoritmo Dijkstra
(Cormen et al., 2001)

En el Algoritmo 1, la línea 1 inicializa los valores d y Π , la línea 2 inicializa el conjunto S en vacío. El algoritmo mantiene la condición invariante $Q = V - S$ para cada iteración del ciclo. La línea 3 inicializa la cola de prioridad Q con todos los vértices en V . Dentro del ciclo se extrae un vértice u de $Q = V - S$, el cual tiene el camino estimado de menor costo, y se añade al conjunto S . Luego se relaja cada arista (u,v) de los vértices adyacentes a u , para actualizar el valor estimado $d[v]$ y el predecesor $\Pi[v]$ si el camino mas corto puede ser mejorado a través de u . Los vértices no se vuelven a insertar en Q y el ciclo itera exactamente $|V|$ veces.

Este algoritmo encuentra el camino de menor costo, en términos de retardo o saltos, desde un nodo fuente a un nodo destino. Sin embargo, los problemas de tráfico de red involucran metas adicionales, sobre todo si se trata de redes inalámbricas, por ejemplo, utilizar el menor ancho de banda o disminuir los caminos perdidos entre nodos.

Enfoque Multiobjetivo

Los problemas de enrutamiento en tráfico de red involucran la optimización simultánea de varios objetivos. En principio, la optimización multiobjetivo es muy diferente de la optimización de un simple objetivo. En optimización de un único objetivo, se intenta obtener el mejor diseño o decisión, la cual es, por lo general, un mínimo local o un máximo local dependiendo del tipo de problema de optimización. Para múltiples objetivos, no existe una única solución, sino un conjunto de alternativas de solución. Estas soluciones son óptimas en el amplio sentido que ninguna otra solución en el espacio de búsqueda es superior a ellas para todos los objetivos considerados, es decir, cuando ningún objetivo puede mejorarse sin degradar algún otro objetivo. Si las metas están en conflicto, ninguna de las soluciones podría ser superior si no se cuenta con información de preferencia (“ranking” de los objetivos). Así, si tal información no esta disponible, se considera útil tener información sobre todas las alternativas posibles para ayudar al gerente en la toma de decisiones.

Al conjunto de estas soluciones óptimas se le conoce como soluciones Pareto óptimas. Su nombre les fue dado en honor al ingeniero y economista Wilfredo Pareto, quien fue el primero en definir un nuevo criterio de optimalidad para los problemas en los que existen múltiples objetivos a optimizar, y existen conflictos al realizar la optimización simultánea de los mismos (Pareto, 1896).

Problema de Optimización Multiobjetivo (POM)

En general, un problema de optimización multiobjetivo esta compuesto por un número de k funciones objetivos que contienen un tupla de h parámetros o variables de decisión y están asociados con un número m de restricciones (ecuaciones o inecuaciones). Matemáticamente, el problema puede expresarse como sigue (Zitzler and Thiele, 1998):

$$\text{Optimizar } y = f(x) = (f_1(x), f_2(x), \dots, f_k(x))$$

$$\text{Sujeto a } e(x) = (e_1(x), e_2(x), \dots, e_m(x)) \geq 0$$

$$x = (x_1, x_2, \dots, x_h) \in X$$

$$y = (y_1, y_2, \dots, y_k) \in Y$$

Siendo x el vector de decisión e y el vector objetivo. El espacio de decisión se denota por X , y el espacio objetivo por Y . Optimizar, dependiendo del problema, puede significar igualmente, minimizar o maximizar. El conjunto de restricciones $e(x) \geq 0$ determina el conjunto de soluciones factibles $X_f \subset X$ y su correspondiente conjunto de vectores objetivos factibles $Y_f \subset Y$.

Conjunto de soluciones factibles

El conjunto de soluciones factibles X_f se define como el conjunto de vectores de decisión x que satisface las restricciones $e(x)$:

$$X_f = \{x \in X \mid e(x) \geq 0\}$$

La región factible en el espacio objetivo compuesta por las imágenes de X_f , se denota por:

$$Y_f = f(X_f) = \bigcup_{x \in X_f} \{y = f(x)\}$$

De acuerdo a estas definiciones, cada solución del POM consta de una h -tupla $x = (x_1, x_2, \dots, x_h)$, que conduce a un vector objetivo $y = (f_1(x), f_2(x), \dots, f_k(x))$, donde cada x debe cumplir con el conjunto de restricciones $e(x) \geq 0$. El problema de optimización consiste en hallar la x que tenga el “mejor valor” de $f(x)$.

A diferencia de la optimización mono-objetivo, las soluciones en un POM no son únicas. Están compuestas por un conjunto de soluciones que representan la mejor

combinación posible de los objetivos. Tales soluciones forman el llamado conjunto Pareto Optimo (PO). Cuando se dibujan los valores de las funciones objetivos correspondientes a las soluciones almacenadas en el conjunto PO, se obtiene el Frente de Pareto del problema. Por ende, al tratar con POMs se precisa de un nuevo concepto de “óptimo”.

Dominancia Pareto

En optimización multiobjetivo, existe una noción diferente respecto a encontrar soluciones "óptimas". Ellas deben garantizar que exista el mejor equilibrio entre los objetivos. Si se tienen dos vectores de decisión u y v tal que $f(u)$ no puede considerarse mejor que $f(v)$ ni $f(v)$ puede considerarse mejor que $f(u)$ ni $f(u) = f(v)$, estamos ante soluciones no comparables entre sí. Para expresar esta situación matemáticamente, las relaciones $=, \leq, y \geq$ se deben extender a POMs. Dados 2 vectores de decisión $u, v \in X$:

$$\begin{aligned} f(u) = f(v) &\Leftrightarrow f_i(u) = f_i(v), \forall i \in \{1, 2, \dots, k\} \\ f(u) \leq f(v) &\Leftrightarrow f_i(u) \leq f_i(v), \forall i \in \{1, 2, \dots, k\} \\ f(u) < f(v) &\Leftrightarrow f(u) \leq f(v) \wedge f(u) \neq f(v) \end{aligned}$$

Las relaciones $\geq, y >$ se definen de manera similar. De esta manera una solución $x^* \in X_f$ es Pareto Óptima si no existe ninguna otra solución $x \in X_f$ que la domine. Se utilizan los símbolos \succ, \prec para denotar la dominancia y \sim para definir no comparabilidad, de la manera siguiente:

Dominancia de Pareto en un contexto de minimización: Para dos vectores de decisión u y v ,

$$u \succ v \Leftrightarrow f(u) < f(v)$$

$$v \succ u \Leftrightarrow f(v) < f(u)$$

$$u \sim v \Leftrightarrow u \not\succ v \wedge v \not\succ u$$

Dominancia de Pareto en un contexto de maximización: Para dos vectores de decisión u y v ,

$$u \succ v \Leftrightarrow f(u) > f(v)$$

$$v \succ u \Leftrightarrow f(v) > f(u)$$

$$u \sim v \Leftrightarrow u \not\succ v \wedge v \not\succ u$$

Alternativamente, $u \triangleright v$ denota que u domina o es igual a v .

No es necesario diferenciar el tipo de optimización a realizar ya que minimizar $f(x)$ es equivalente a maximizar $-f(x)$. Así, en un mismo problema se puede encontrar un objetivo a ser maximizado, mientras que otro puede ser minimizado.

Optimalidad Pareto

Dado un vector de decisión $x \in X_f$, se dice que x es no dominado respecto a un conjunto $V \subseteq X_f$ si y solo si:

$$\forall v \in V : (x \triangleright v \vee x \sim v)$$

En caso que x sea no dominado respecto a todo el conjunto X_f , y solo en ese caso, se dice que x es una *solución Pareto óptima*.

Conjunto Pareto Óptimo

Dado el conjunto de vectores de decisión factibles X_f , se denomina conjunto Pareto óptimo X_{PO} al conjunto de vectores de decisión no dominados de X_f , es decir:

$$X_{PO} = \{x \in X_f \mid x \text{ es no-dominado con respecto a } X_f\}$$

El correspondiente conjunto de vectores objetivos $Y_{PO} = f(X_{PO})$ constituye el *frente Pareto Óptimo*.

Toma de Decisiones

El campo de investigación de problemas de decisión con varios objetivos en conflicto (o metas en conflicto) es conocido como Toma de Decisiones con Múltiples Criterios (TDMC). Cubre tanto problemas discretos (con un conjunto finito de alternativas de solución) como continuos. Para resolver problemas multiobjetivo, frecuentemente es necesario contar con información de preferencia para obtener una única solución final (Deb et al., 2008). Hay varias formas de incorporar las preferencias del usuario durante la búsqueda:

- *Métodos a Priori*: El responsable de las decisiones indica información de preferencia en los diferentes criterios antes de comenzar el proceso de optimización para obtener una única solución. Este método suele ser muy rápido, pero debe considerarse el tiempo necesario para modelar dichas preferencias. Además, el tomador de decisiones puede sentirse insatisfecho con la solución encontrada y volver a repetir en otro proceso de optimización un compromiso diferente entre los objetivos.

- *Métodos Progresivos*: El responsable de las decisiones participa durante el proceso de optimización expresando sus preferencias para guiar la búsqueda. Aunque aseguran que la información de importancia se toma en cuenta con precisión, estos métodos requieren de mucha interacción durante la búsqueda.
- *Enfoques a Posteriori*: En esta tercera categoría, las preferencias en los objetivos no necesitan ser modeladas antes de tiempo. Una vez terminado el proceso de optimización, el responsable de las decisiones elegirá entre todas las soluciones generadas (que idealmente son todas del conjunto Pareto óptimo), la que considere más apropiada. Sin embargo, requiere gran esfuerzo computacional.

Cada uno de estos enfoques tienen ventajas y desventajas y elegir alguno depende del tomador de decisiones y la naturaleza del problema. En este trabajo se adoptará el enfoque a posteriori por ser tradicionalmente usado en las metaheurísticas multiobjetivo.

El Método de Suma con Pesos

Este método combina las diferentes funciones objetivo en una sola función F , de la siguiente manera:

Optimizar

$$F = \sum_{j=1}^k w_j f_j(x)$$

Donde $x \in X_f$, y w_j es el peso usado para ponderar la j -ésima función objetivo, usualmente:

$$\sum_{j=1}^k w_j = 1, w_j > 0$$

Las soluciones de la función F son muy cercanas al frente de Pareto óptimo y, más aún es Pareto óptimo si hacemos $w_i > 0, \forall i = 1, \dots, k$ o si la solución es única (Deb et al., 2008). El procedimiento es sencillo: se escoge una combinación de pesos y se optimiza la función F para obtener una solución óptima. Otras soluciones surgen a partir de optimizaciones realizadas sobre una combinación diferente de pesos y se repite hasta que se cumpla algún criterio de parada preestablecido. En este contexto, optimizar implica maximizar todas las funciones objetivo, o minimizarlas. Es decir, no admite la maximización de algunos objetivos y la minimización de otros.

El método de suma con pesos puede ser usado como un método a posteriori ya que diferentes pesos son utilizados para generar múltiples soluciones Pareto-óptimas y luego el responsable de las decisiones seleccionará la más satisfactoria. Alternativamente, se pueden especificar los pesos en el caso de que el método se use a priori.

Es importante que la optimización multiobjetivo genere soluciones Pareto-óptimas y que puedan ser encontradas. En este aspecto, el método de suma con pesos tiene un serio inconveniente. Puede encontrar una solución Pareto-óptima alterando los pesos, solo si el problema es convexo. Esto se debe a que, al alterar los pesos, el método puede saltar de un vértice a otro ignorando las soluciones intermedias (Deb, 1999). Este es un aspecto serio e importante, ya que no siempre es fácil comprobar la convexidad en aplicaciones reales si el problema se basa, por ejemplo, en algún modelo de simulación o la solución en sistemas de ecuaciones de derivadas parciales.

Aunque no se adecuan a la naturaleza del problema como tal, la ventaja principal del método de suma con pesos es que permite la utilización de algoritmos de optimización mono-objetivo, bien conocidos y de probada eficacia, en problemas reales de gran tamaño. Sin embargo, este método cuenta con las siguientes limitaciones (Deb, 1999):

- a) El algoritmo de optimización se debe aplicar varias veces para encontrar múltiples soluciones Pareto-óptimas. Como cada corrida es independiente de las demás, generalmente no se obtiene un efecto sinérgico. Por tanto, delinear el frente Pareto óptimo resulta computacionalmente muy caro.
- b) Requiere conocimiento previo del problema a resolver y es sensible a los pesos utilizados.
- c) Es sensible a la forma del frente Pareto (problemas con curvas cóncavas).
- d) La variación entre las diferentes soluciones encontradas depende de la eficiencia del optimizador de un solo objetivo. Podría darse el caso de encontrar siempre la misma solución o soluciones muy parecidas, en corridas múltiples.
- e) Como los optimizadores de un solo objetivo no son eficientes en búsquedas de universos discretos, tampoco serán eficientes para optimizaciones multiobjetivo en espacios discretos.

Investigaciones publicadas por Deb (1999) han demostrado que todas estas dificultades pueden ser superadas con la utilización de algoritmos evolutivos. Las características de estos algoritmos hacen posible que:

- a) Se manejen espacios de búsqueda de casi cualquier tamaño y características;
- b) Debido a su paralelismo intrínseco, se puedan generar múltiples soluciones en una sola corrida, permitiendo un efecto sinérgico.

Algoritmos Evolutivos

Los Algoritmos Evolutivos de optimización se diferencian de los métodos clásicos de optimización en los siguientes aspectos (Goldberg, 1989):

- Generalmente no utilizan información del gradiente durante el proceso de búsqueda y no pueden competir en problemas de programación convexa, lineal o cuadrática con los algoritmos de optimización basados en gradiente.

- Debido a su enfoque poblacional (procesamiento paralelo), generan más de una solución en cada iteración presentando las siguientes ventajas: búsquedas computacionalmente rápidas, facilidad para encontrar soluciones en problemas multimodales y habilidad para normalizar las variables de decisión dentro de la población evolutiva usando sus valores mínimos y máximos. Sin embargo, trabajar con enfoques poblacionales tiene costos adicionales de procesamiento y memoria en cada iteración.
- Usan operadores estocásticos a diferencia de los operadores determinísticos presentes en los métodos clásicos de optimización. Estos tienden a lograr las salidas deseadas usando distribuciones de probabilidad sesgadas. Por tanto, permite que se generen múltiples óptimos con igual posibilidad proporcionando una perspectiva global en la búsqueda.

El término Algoritmo Evolutivo (AE) se refiere a técnicas de búsqueda y optimización inspiradas en el modelo de evolución propuesto por Charles Darwin (1985). En la naturaleza los individuos se caracterizan mediante cadenas de material genético que se denominan cromosomas. En los cromosomas se halla codificada toda la información relativa a un individuo y a sus tendencias. El cromosoma es una cadena de símbolos llamados genes. Cada individuo posee un nivel de adaptación al medio que lo dota de mayor capacidad de sobrevivencia y generación de descendencia. Tal nivel de adaptabilidad está ligado a las características codificadas en sus cromosomas.

Como el material genético puede transmitirse de padres a hijos al ocurrir el apareamiento, los hijos resultantes poseen cadenas de cromosomas parecidas a las de sus padres y combinan las características de los mismos. Por lo tanto, si padres con buenas características se cruzan, posiblemente generarán hijos igualmente buenos o incluso mejores (Goldberg, 1989).

Un algoritmo evolutivo comienza la búsqueda con una población P de soluciones (cromosomas), generalmente inicializadas con valores aleatorios acotados entre un máximo y un mínimo. Luego, el procedimiento entra en una operación iterativa al actualizar la población para crear una nueva población usando sus cuatro operadores principales: selección, cruce, mutación y elitismo. El algoritmo se detiene cuando se cumplen los criterios previamente establecidos.

```

Inicio
   $t = 0;$ 
  Inicializacion( $P_t$ );
  Repita {
    Evaluación( $P_t$ );
     $P'_t = \text{Selección}(P_t)$ 
     $P''_t = \text{Variación}(P'_t);$ 
     $P_{t+1} = \text{Elitismo}(P_t, P''_t);$ 
     $t = t + 1;$ 
  } Hasta Parada( $P_t, P''_t$ );
Fin

```

Algoritmo 2: *Procedimiento de Optimización Evolutiva*
(Goldberg, 1989)

En el Algoritmo 2, El proceso de evaluación de la población consiste en calcular el valor de la función objetivo para cada miembro de la población y así establecer una clasificación de las soluciones (desde la mejor hasta la peor) dentro de la población. Para problemas de optimización multiobjetivo, este orden esta dado por el principio de no-dominación mencionado anteriormente. Luego, el operador de selección elige con mayor probabilidad a las mejores soluciones para que formen parte del pool de apareo. El procedimiento de variación reúne a los operadores de cruce, mutación, entre otros, para generar cambios en la población seleccionada. Su propósito es tomar al azar soluciones padres del pool de apareo y crear una o mas

soluciones hijas intercambiando información de las soluciones padres. Entonces, cada solución hija es perturbada dentro de su cromosoma por mutación para introducir diversidad en el espacio de búsqueda. El operador de elitismo combina la población anterior con la nueva para resguardar las mejores soluciones y asegurar que el algoritmo tenga un comportamiento monolítico no degradado. Para finalizar la iteración se puede utilizar un número de generaciones o una tasa de convergencia como criterio de parada.

Los operadores descritos reciben el nombre de operadores de búsqueda u operadores genéticos. La reproducción enfoca la atención en los individuos con alta adaptabilidad, y de esta manera *explota* la información disponible sobre la adaptación del individuo al medio ambiente. El cruce y la mutación modifican de alguna manera a los individuos y proveen así de heurísticas para la *exploración* del espacio de búsqueda. Por ello se dice que los AEs utilizan conceptos de explotación y exploración. A pesar de ser simplistas desde el punto de vista de la biología, estos algoritmos son suficientemente complejos como para proveer mecanismos de búsqueda robustos y que se adaptan a gran variedad de problemas (Goldberg, 1989).

Algoritmos Genéticos

Jhon Holland alrededor de 1975, describió un marco general para visualizar todos los sistemas adaptativos naturales o artificiales, y expuso cómo el proceso evolutivo puede aplicarse a los sistemas artificiales. Cualquier problema de adaptación puede ser formulado en términos genéticos. Una vez formulados, pueden ser resueltos por lo que ahora se conoce como "Algoritmos Genéticos".

Los Algoritmos Genéticos (AG) se utilizan en varios ámbitos, principalmente para búsquedas y optimizaciones. En la práctica se implementa el algoritmo escogiendo una codificación para las posibles soluciones del problema. La codificación se realiza mediante cadenas de bits, números o caracteres para

representar a los cromosomas. Luego, las operaciones de cruce y mutación se aplican de manera muy sencilla mediante funciones de manipulación de vectores. A pesar de que se han realizado numerosas investigaciones (Heitkoetter, 2000) sobre codificación usando cadenas de longitud variable y aún otras estructuras, gran parte del trabajo con algoritmos genéticos se enfoca en cadenas de bits de longitud fija. Justamente el hecho de utilizar cadenas de longitud fija y la necesidad que existe de codificar las posibles soluciones son las dos características cruciales que diferencian a los algoritmos genéticos de la programación genética, que no posee representación de longitud fija y que normalmente no utiliza una codificación del problema y sus soluciones.

El ciclo de trabajo de un AG (también conocido como ciclo generacional) generalmente es el siguiente: calcular la adaptación de los individuos en la población, crear una nueva población mediante selección, cruce y mutación, y finalmente descartar la población vieja y seguir iterando utilizando la población recién creada. A cada iteración de este ciclo se le conoce como una generación. Cabe observar que no hay una razón teórica para que este sea el modelo de implementación. De hecho este comportamiento puntual no se observa como tal en la naturaleza. Sin embargo el modelo sigue siendo válido y conveniente (Goldberg, 1989).

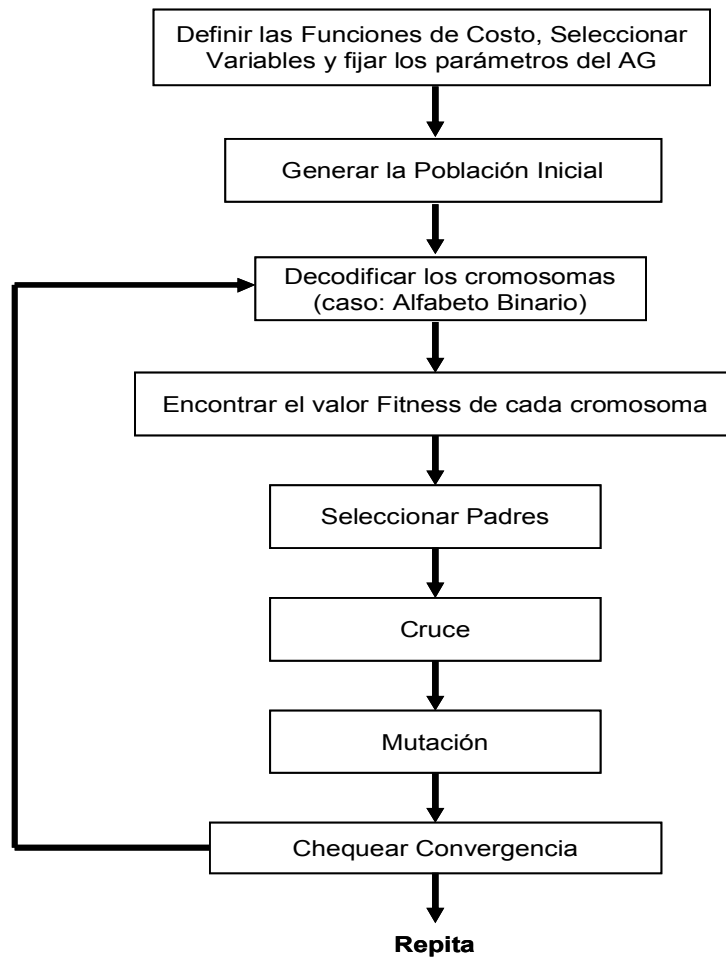


Figura 2: *Funcionamiento general de Un Algoritmo Genético.*
(Haupt, 1998)

La Figura 2 (Haupt, 1998), explica a grandes rasgos el funcionamiento de un Algoritmo Genético. Entre los aspectos más resaltantes se debe evaluar la adaptabilidad (Fitness) de cada uno de los individuos. El operador de selección, simula el proceso de selección natural donde el más fuerte tiene mayor capacidad de supervivencia. En el AG la capacidad de supervivencia de un individuo está ligada a su valor de adaptabilidad y de acuerdo a este valor, permitir a los “mejores” individuos reproducirse. El operador de selección puede aplicarse de diversas maneras. A continuación se describen dos implementaciones muy populares:

- **Selección por ruleta:** Se podría comprender imaginando una ruleta en la que el número de partes en que se divide la misma es igual a la cantidad de individuos de la población, siendo el tamaño de cada parte proporcional a la adaptación de cada individuo. Se espera que al girar la ruleta varias veces, se obtenga mayor cantidad de individuos con alta adaptabilidad. Un número aleatorio determina el cromosoma seleccionado.
- **Selección por torneos:** Simula la selección que ocurre en la naturaleza ya que elige un grupo de individuos del pool de apareo y aquellos con mejor adaptabilidad se convierten en padres de la próxima generación. La cantidad de individuos que se escogen para la competencia se fija de antemano y permanece constante en la implementación tradicional. Este tipo de selección trabaja mejor para poblaciones de gran tamaño que métodos basados en clasificación, ya que requieren tiempo de procesamiento adicional.

El cruce es la creación de uno o más descendientes de los padres seleccionados en el proceso anterior. El genotipo de la población está limitado por los miembros de dicha población. La forma más común de cruce es involucrar a dos padres para producir dos hijos. La descendencia contiene en sus cromosomas porciones de código de ambos padres. Para ello se pueden elegir uno o más puntos de cruce:

- **Cruce en un punto:** Un punto es seleccionado aleatoriamente entre el primero y el último gen del cromosoma padre. El primer padre pasa su código a la izquierda del punto de cruce a su primer hijo, al igual que el segundo padre lo hace con su segundo hijo. Luego el código a la derecha del punto de cruce del primer padre va al segundo hijo mientras que el segundo padre lo transmite a su primer hijo.
- **Cruce en dos puntos:** Se escogen aleatoriamente dos puntos de cruce. El primer punto de cruce define el gen a partir del cual se hará el intercambio genético, y el

segundo define la longitud del segmento (número de genes) a intercambiar. Su intención es introducir mayor variedad en la descendencia en la búsqueda de mejores soluciones.

El operador de cruce representa una forma de búsqueda local, en las inmediaciones del espacio de búsqueda que rodea a los padres. Por su parte, el proceso de mutación es básicamente una búsqueda aleatoria. Las mutaciones alteran un cierto porcentaje de genes en la lista de cromosomas. Ella introduce hitos en la población actual y ayuda al AG a converger mas rápidamente explorando nuevos espacios de soluciones. Incrementar el número de mutaciones incrementa la libertad del algoritmo de buscar fuera de la actual región de búsqueda. Sin embargo, tiende a distraer al algoritmo en la convergencia.

Durante el proceso de mutación se selecciona aleatoriamente una posición específica dentro de cromosoma del individuo y se cambia el valor contenido en dicha posición. Dado que en la naturaleza la probabilidad de que ocurra una mutación es pequeña, los AGs tratan de representar lo mismo asignando un valor de probabilidad muy bajo al operador de mutación (Goldberg, 1989).

Algoritmos Evolutivos en Optimización Multiobjetivo (AEMOs)

Los AEs resultan interesantes pues están especialmente dotados para lidiar con las dificultades propuestas por los POMs. Esto se debe a que pueden obtener un conjunto entero de soluciones luego de una corrida simple y no presentan otras limitaciones propias de las técnicas tradicionales. Incluso, algunos investigadores han sugerido que los AEs se comportan mejor que otras técnicas de búsqueda ciega para cierto tipo de problemas (Fonseca and Fleming, 1995). Además, actualmente se cuenta con pocas alternativas válidas en cuanto a otros posibles métodos de solución (Horn, 1997) y han surgido numerosas publicaciones (Coello, 2000) que han dado

lugar al nacimiento de todo un nuevo campo de investigación: los AEMOs, algoritmos evolutivos aplicados a optimización multiobjetivo.

En particular, esta investigación se ha enfocado en técnicas que han demostrado ser capaces de explorar espacios de búsquedas prácticamente ilimitados y altamente complejos y generar aproximaciones lo más fieles posibles al conjunto Pareto óptimo. Los AEMOs constituyen una alternativa general y robusta (Zitzler and Thiele, 1998) debido a su enfoque poblacional y que no requieren mayor información sobre el espacio de búsqueda; basta con especificar los objetivos y la manera de calcularlos.

Su popularidad se debe principalmente a que:

- a) resultan promisorios para la integración de los aspectos de búsqueda y de toma de decisiones que plantean los POMs;
- b) pueden realizar búsquedas aún en espacios ilimitados y altamente complejos;
- c) tienen la habilidad de mantener toda una población de soluciones;
- d) existen pocas alternativas –aparte de ellos– para el tratamiento adecuado de POMs.

Por tales motivos, en el presente trabajo se empleara un algoritmo genético basado en SPEA para tratar el problema de enrutamiento en redes de sensores inalámbricos.

Algoritmo Evolutivo SPEA (Strength Pareto Evolutionary Algorithm)

Ha sido catalogado como uno de los algoritmos más promisorios debido a su capacidad de obtener buenos resultados bajo pruebas exhaustivas en problemas de prueba cuidadosamente seleccionados (Zitzler and Thiele, 1998). SPEA usa una mezcla de técnicas previamente usadas y otras recién establecidas para encontrar

soluciones Pareto óptimas diferentes y en paralelo. Las características que comparte con otros algoritmos previamente propuestos son:

1. Almacena el frente de Pareto óptimo externamente. De esta manera implementa elitismo, que consiste en seleccionar las mejores soluciones de la generación actual y copiarlas en la población de la siguiente generación;
2. Usa el concepto de dominancia Pareto para la asignación de adaptabilidad a los individuos;
3. Realiza *clustering* (Morse, 1980) para reducir el número de soluciones no dominadas que tiene almacenadas, sin destruir las características del frente Pareto delineado hasta el momento.

Por otro lado, SPEA presenta, de manera exclusiva, los siguientes conceptos:

1. Combina las tres características arriba mencionadas en un único algoritmo, las cuales habían sido propuestas de forma separada;
2. La adaptación de un individuo se determina a partir de las soluciones que se encuentran almacenadas en el conjunto externo de Pareto solamente; de este modo, es irrelevante que los miembros de la población general se dominen unos a otros;
3. Todas las soluciones que forman parte del conjunto externo de soluciones no dominadas participan en la selección;
4. Se sugiere un nuevo método para inducir la formación de nichos dentro de la población y así preservar la diversidad genética; este método está basado en el concepto de dominancia Pareto.

Todos los individuos no dominados en la población son copiados al conjunto externo de Pareto, y las soluciones dominadas por estos individuos son removidas. Si el número de soluciones de Pareto exceden un máximo dado, se realiza reducción del conjunto vía “clustering”. Se une el conjunto de Pareto y la población para asignar

adaptabilidad a los individuos. Luego se hace torneo binario para llenar el pool de apareo. Finalmente, se aplica cruce y mutación a la población.

Una de las particularidades del SPEA es su forma de asignar el valor de adaptación a cada individuo. El proceso se realiza de modo que los individuos dominantes y más representativos tengan mejores valores de adaptación. El conjunto externo de Pareto es usado para evaluar los individuos en la población actual, inspirado por trabajos en el campo de sistemas inmunes para resolver problemas de adaptación (población de anticuerpos y antígenos), donde los individuos son evaluados dependiendo del número de puntos de Pareto que los "cubren" (Zitzler and Thiele, 1998) y de esta manera, descubrir nuevas soluciones no dominadas. Este valor de "cobertura" representa el valor de adaptación para un individuo particular.

A cada solución en el Conjunto de Pareto se le asigna un valor real $s \in [0,1)$, llamado "strength" (fuerza); s es proporcional al número de miembros de la población cubierta. La fuerza de una solución Pareto es al mismo tiempo su valor de adaptación o "fitness". Luego, los individuos de la población actual son clasificados con respecto a los valores de fuerza calculados. Es decir, por cada individuo se suman las fuerzas de todas las soluciones de Pareto que lo "cubren". Se suma uno (1) al valor resultante para garantizar que las soluciones de Pareto tengan mayor probabilidad de reproducirse, y se obtiene el valor de adaptación o "fitness" f , donde $f \in [1,N)$ y N denota el tamaño de la población.

Si todas las soluciones de Pareto tienen igual fuerza, el fitness de un individuo está totalmente determinado por el número de puntos de Pareto que lo cubren. Sin embargo, en el caso de que la población esté desbalanceada, las fuerzas actúan así: El punto de Pareto más fuerte es el que cubre al menor número de individuos. Para ver el efecto de este método de clasificación, se puede dividir el espacio objetivo en subespacios, los cuales están definidos por el conjunto de Pareto; los individuos que pertenecen a un subespacio están exactamente cubiertos por los mismos puntos de

Pareto. La meta es distribuir la población uniformemente cerca del frente de Pareto Óptimo tal que todas las soluciones de Pareto cubran la misma cantidad de individuos.

SPEA trata de distribuir los individuos de manera que:

- a) Los subespacios igualmente rellenos compriman el mismo número de soluciones.
- b) los subespacios con mejores fitness contengan más individuos, en contraste a los puntos de Pareto mas desolados, los cuales se eliminan, ya que no reflejan el espíritu de la población.

Finalmente, el esquema de selección usado en este estudio es selección por torneo binario. Como se menciona arriba, ambas poblaciones y el conjunto externo de Pareto son considerados para reproducción: El fitness más pequeño de un individuo tiene la más alta probabilidad de ser seleccionado.

Reducción del Conjunto Externo de Pareto

En ciertos problemas, el conjunto de Pareto-óptimo puede ser extremadamente grande o contener un número infinito de soluciones. En este caso, es conveniente podar las soluciones no-dominadas por las siguientes razones:

1. Presentar todas las soluciones de Pareto encontradas es poco útil cuando el número excede límites razonables, independientemente si se desea explorar el espacio objetivo o alcanzar una solución final.
2. Cuando el Frente de Pareto es continuo, físicamente no es posible (o deseable) guardar todas las soluciones no dominadas en un medio de almacenamiento principal o secundario.

3. El tamaño del conjunto externo de Pareto puede incrementarse a múltiples tamaños de población. En esta situación, prematuramente todas las soluciones de Pareto serán clasificadas por igual, reduciendo la presión selectiva, y en consecuencia el proceso de optimización se hace substancialmente lento.
4. Como se menciona antes, el espacio objetivo cubierto por el conjunto externo de Pareto, puede ser dividido en subespacios; la efectividad del mecanismo de nicho propuesto depende de la granularidad de estos subespacios. En un escenario óptimo todos los subespacios son iguales en tamaño tendiendo a una distribución uniforme de los individuos. Sin embargo, cuando el conjunto de Pareto no esta bien distribuido, puede ocurrir muchas variaciones en el tamaño de los subespacios. Como consecuencia, el método de asignación de fitness es influenciado hacia ciertas regiones del espacio de búsqueda, ocasionando una distribución desbalanceada de la población.

En SPEA, la técnica de reducción del Frente de Pareto aplicando análisis por cluster, divide una colección de p elementos en q grupos de elementos relativamente homogéneos, donde $q < p$. Dependiendo del desempeño del algoritmo dos formas de agrupamiento o “clustering” se consideran: Clustering Directo y Clustering Jerárquico. Mientras el primero ordena los p elementos en q grupos en un paso, el último trabaja iterativamente al unir grupos adyacentes hasta que se obtenga el número de grupos requerido.

Al comienzo cada elemento del conjunto de Pareto original forma un “cluster” básico. Luego en cada paso dos grupos son combinados en un “cluster” más grande hasta que se alcance el número de grupos deseado. Los cluster a combinarse, son seleccionados de acuerdo al criterio de los vecinos mas cercanos donde la distancia entre dos cluster esta dada como la distancia promedio entre pares de individuos a través de los dos grupos. Finalmente, cuando el procedimiento de división finaliza, el

Frente de Pareto reducido se forma al seleccionar el individuo representativo de cada cluster. Se considera el *Centroide* (el punto con la mínima distancia al resto de puntos en el cluster) como una solución representativa.

La Métrica Spacing (S)

En un problema de optimización multiobjetivo además de perseguir la convergencia al conjunto Pareto óptimo, también es deseable mantener la diversidad en las soluciones obtenidas. Para tratar formalmente la uniformidad en la distribución de las soluciones no dominadas Pnd , se utiliza la métrica “Spacing” (S) definida formalmente por Deb (2001):

$$S = \sqrt{\frac{1}{|Pnd|} \sum_{i=1}^{|Pnd|} (d_i - \bar{d})^2}$$

Donde,

$$d_i = \min_{k \in Pnd \wedge k \neq i} \sum_{m=1}^M |f_m^i - f_m^k| \quad \text{y} \quad \bar{d} = \sum_{i=1}^{|Pnd|} \frac{d_i}{|Pnd|}$$

La métrica S se calcula como la distancia relativa entre soluciones consecutivas en el conjunto no dominado ya que mide las desviaciones típicas de diferentes valores d_i . Un valor muy bajo -cercano a cero- indica que las soluciones están uniformemente distribuidas.

CAPITULO III

MARCO METODOLOGICO

Tipo de Investigación

El estudio se apoya en una investigación bajo la modalidad de proyectos tal como lo especifica el “Manual para la Presentación de Trabajo de Grado” (UCLA, 2002); ya que desarrollar un algoritmo de enrutamiento basado en SPEA para redes de sensores inalámbricos, permite el diseño de un proyecto factible para la solución de un problema de tipo práctico.

Adicionalmente se encuentra apoyado en la investigación de campo de carácter documental. Según el “Manual para la Presentación de Trabajo de Grado” de la Universidad Centroccidental “Lisandro Alvarado” (UCLA, 2002) se entiende por investigación de campo:

La aplicación del método científico en el tratamiento de un sistema de variables y sus relaciones, las cuales conducen a conclusiones y al enriquecimiento de un campo del conocimiento o disciplina inherente a la especialidad, con la sustentación de los experimentos y observaciones realizadas. (p. 4).

El algoritmo evolucionará en un ambiente de configuración de red simulado. Con los resultados obtenidos se evaluará la convergencia del conjunto de rutas de Pareto, utilizando como referencia un número de generaciones determinado.

Estudio de Factibilidad

Factibilidad Técnica

El desarrollo del algoritmo propuesto requiere de recursos de hardware y software. Entre los requisitos mínimos de hardware se encuentran: Procesador Pentium IV, 3 GB de RAM, 120 GB de disco duro, monitor a color, ratón, teclado, unidad de CD/RW.

En cuanto al software, el lenguaje de programación a emplear es c++ utilizando el entorno Eclipse (compilador g++ cygwin libre). Su principal ventaja es su uso multiplataforma, permitiendo el desarrollo del algoritmo en sistemas operativos abiertos como Linux, o su implantación en el sistemas propietarios como Windows, Mac, entre otros.

Factibilidad Económica

Actualmente se dispone de un Servidor Intel Xeon en la Unidad de Inteligencia Artificial de Decanato de Ciencias de la UCLA, con características superiores a las requeridas en el estudio de factibilidad técnica. Por tanto, los experimentos y análisis de resultados se ejecutarán sin inconvenientes en dicho equipo, utilizando una licencia Windows Server 2003 disponible para uso académico dentro de la universidad.

Procedimiento de la Investigación

En cuanto al procedimiento de la Investigación, se continuará haciendo una revisión bibliográfica para el logro de los objetivos. En primer lugar, se diseñará un mecanismo de configuración de la red de sensores inalámbricos para definir un

ambiente de prueba. Esta simulación será insumo para el algoritmo genético multiobjetivo.

Como la evaluación de las funciones objetivos para un individuo dentro de la población genética es distinta para el camino inter-cluster e intra-cluster, se decidirá un mecanismo que unifique ambos valores en un único resultado.

Luego, tomando como base el motor evolutivo SPEA, se construirá el algoritmo que medirá todas las características presentes en la arquitectura de la red e irá evolucionando progresivamente en búsqueda de una población con soluciones cercanas al conjunto óptimo (Frente de Pareto).

Finalmente, se consultarán las teorías que hasta la fecha se han publicado sobre las medidas de desempeño para evaluar el rendimiento del algoritmo y evaluar su convergencia utilizando un número de generaciones determinado.

CAPÍTULO IV

DISEÑO E INGENIERÍA DE LA PROPUESTA

Formulación Matemática de los Objetivos a Optimizar

En el presente trabajo, la red de sensores inalámbricos está formada por clústeres, cada uno de los cuales es modelado como un grafo totalmente conexo $G = (V, E)$, donde V es el conjunto de vértices y E el conjunto de aristas. Los vértices del grafo representan los sensores de la red, y las aristas representan los enlaces entre los nodos. Dentro del grafo se distingue un nodo especial o nodo cabecera que tiene mayores prestaciones en cuanto a capacidad de procesamiento y vida de batería. A su vez todos los nodos “cabeceras de clúster” forman un grafo completamente conectado: pueden comunicarse unos con otros y con la estación central, donde llegará la data recolectada por el agente. Sea:

Ta_k : Tiempo de adquisición de data para el nodo k.

Tp_k : Tiempo de procesamiento de Data para el nodo k.

H_k : Frecuencia operativa del sensor k; H_k^2 determina su nivel de potencia.

Pt_k : Poder de transmisión del nodo k (cuánto gasta en energía para transmitir).

Tm : Tiempo de Transmisión de Mensajes.

$E(P)$: Consumo Total de Energía.

d_{ij} : Distancia euclídea entre las coordenadas de los sensores i y j.

Gt_j : Ganancia de transmisión del sensor j.

Gr_j : Ganancia de recepción del sensor i.

L : Longitud de honda = Velocidad/Frecuencia.

β : Factor de pérdidas de filtros, líneas de transmisión, Acoples Amplificadores etc. Por lo general $\beta \geq 1$.

PL_k : Pérdida del camino asociado con una arista k .

E : Conjunto de todas las aristas de la ruta.

d_k : Distancia euclídea entre las coordenadas del objetivo y las coordenadas del sensor k .

p : Exponente de decaimiento de la señal, toma valores entre 2 y 3.

λ : Constante ajustable.

$PL(P)$: Camino Total Perdido de una ruta P .

$e_k(u)$: Cantidad de Energía capturada por un sensor k .

$DE(P)$: Detección Total de Energía.

Usando las notaciones mencionadas, el problema de enrutamiento en una red de sensores inalámbricos puede ser definido como un POM que trata de hallar un conjunto de rutas óptimas resolviendo las siguientes funciones objetivos (Rajagopalan et al., 2005):

1) Minimización del Consumo total de Energía

Como se trata de una red de sensores inalámbricos heterogéneos, todos los sensores cuentan con una capacidad de batería limitada de diferentes niveles, en comparación a otros sensores. Cada sensor consume cierta energía en la adquisición, procesamiento y transmisión de datos. Por tanto, el consumo de energía de la red depende de la capacidad de un sensor y su funcionalidad. Los mensajes transmitidos entre sensores están incluidos en el código del agente móvil de tamaño M bits y la data de tamaño K bits, la cual va creciendo en la medida que se va recolectando. El tiempo de transmisión de mensajes sobre un canal inalámbrico de ancho de banda B esta dado por:

$$Tm = (M + K) / B$$

El consumo de energía de un camino P es la suma de la energía gastada por cada sensor a través del camino (Rajagopalan et al., 2005). Si $n_0, n_1, n_2, \dots, n_l$ denota la secuencia de nodos a través de un camino P , entonces el total de energía consumida $E(P)$ esta dada por:

$$E(P) = \sum_{k=0}^l ((Ta_k + Tp_k) * H_k^2) + (Pt_k * Tm) \quad (1)$$

2) Minimización de la Pérdida Total de Caminos

Los enlaces de comunicación inalámbrica necesitan ser establecidos entre los sensores vecinos a medida que el agente móvil atraviese una ruta. El nivel de Señal recibida podría no ser aceptable si esta por debajo de cierto umbral, lo cual produce una pérdida de camino. La pérdida de camino representa la atenuación de la señal debido a la propagación del espacio libre, y debería ser minimizado para garantizar una comunicación confiable.

La pérdida total de caminos a través de una ruta es la suma de las pérdidas de caminos asociados con cada enlace a través del camino (Rappaport, 1996). La pérdida de camino asociado con un simple enlace $PL(d_{ij})$ es calculada -utilizando el modelo de propagación en espacio libre de Friis (1946)- como la potencia entre la energía Pt_i transmitida por el sensor i y la energía Pr_{ji} recibida por el sensor j (medida en dB):

$$PL(d_{ij}) = 10 * (\log(Pt_i / Pr_{ji}))$$

A su vez la energía recibida por el sensor j , proveniente del sensor i viene dada por la siguiente fórmula:

$$Pr_{ij} = \left(Pt_j * Gt_j * Gr_i * L^2 \right) / ((4 * \pi * d_{ji})^2 * \beta)$$

Finalmente, la pérdida total de caminos de una ruta P es calculada por:

$$PL(P) = \sum(PL_k), k \in E \quad (2)$$

3) Maximización de la Exactitud en la Detección

Para todos los sensores de la red, lograr la mayor calidad en la medición de la señal recibida, es una meta primordial para que el agente pueda fusionar data confiable. Datos corruptos no podrán ser procesados por la estación central y el agente tendría un mal desempeño.

Cada sensor k detecta cierta cantidad de energía $e_k(u)$, emitida por una fuente del ambiente en adelante llamada *Objetivo*. Si K_0 es la energía emitida por un objetivo en las coordenadas $u = (x_i, y_i)$, la señal de energía e_k medida por un sensor es:

$$e_k(u) = K_0 / (1 + \lambda * d_k^p)$$

En un clúster con s nodos, cada sensor tiene $s-1$ vecinos y todos reciben valores de los mismos objetivos. Según el modelo bizantino de sensores defectuosos (Rajagopalan et al., 2005), los sensores de la red intercambian sus valores de energía local y los fusionan al encontrar el promedio.

Así, la energía representativa E_i del i -ésimo sensor, viene dada por la siguiente fórmula:

$$E_i = \sum_{k=1}^m e_k(u) / m$$

E_i representa la media de los $e_k(u)$ de m vecinos escogidos aleatoriamente en el intervalo $[3, s-1]$.

Dentro de un clúster, la energía representativa del nodo cabecera E_c se calcula como la media de todos los valores de energía detectados por los sensores de ese clúster:

$$E_c = \sum_{i=1}^s E_i / s$$

Para calcular la Detección Total de Energía, la estación central decide la secuencia de clústeres que el agente móvil deberá visitar basado en la energía representativa de los n nodos cabecera, la cual viene dada por la sumatoria de la energía de la señal detectada a través del camino P :

$$DE(P) = \sum_{c=1}^n E_c \quad (3)$$

De esta manera, el problema de optimización tiene varios objetivos en conflicto y el frente de Pareto óptimo debe alcanzarse con la mayor exactitud.

Simulación de la Red

Siguiendo el modelo descrito por Rajagopalan et al. (2005), el ambiente de simulación de la red cuenta con las siguientes características:

- Una red de sensores conectados a través de enlaces de comunicación inalámbrica, la cual está formada por clústeres, y cada clúster compuesto por sensores.

- Los sensores dentro de cada clúster forman un grafo totalmente conexo, comunicándose unos con otros y midiendo la misma información del ambiente.
- A su vez todos los nodos “cabeceras de clúster” forman un grafo totalmente conexo, ya que pueden comunicarse unos con otros y con la estación central, donde finalmente llegará la data medida por cada uno de los sensores con el fin de ser procesada.

Para ello, el algoritmo solicita una serie de parámetros de entrada entre los que se encuentran:

Exp : Número de Experimentos.

Gen : Número de Generaciones.

Tampob : Tamaño de la población.

Par : Tamaño Máximo del Frente de Pareto.

M : Tamaño del Agente Móvil (bits).

K : Tamaño promedio de data (bits).

Gt : Ganancia en la Transmisión para todos los sensores.

Gr : Ganancia en la Recepción para todos los sensores.

B : Ancho de Banda (bps).

N : Nro. de Clústeres.

S : Nro. de Sensores internos por Clúster.

Obj : Nro. De objetivos por Clúster.

P_c : Probabilidad de Cruce.

P_m : Probabilidad de Mutación

Adicionalmente, solicita las coordenadas mínimas y máximas donde se distribuirá cada clúster ($[Xmin - Xmax]$, $[Ymin - Ymax]$) y las coordenadas de los objetivos a medir en el ambiente. Esto con el fin de distribuir en el plano cartesiano la arquitectura de la red.

El algoritmo inicializa una matriz de sensores de dimensiones $n \times (s + 1)$, donde n representa el número de clústeres y s el número de sensores por clúster (incluyendo el nodo cabecera, ubicado en la columna cero (0) de cada cluster). Se definen para cada sensor una serie de valores asignados aleatoriamente:

- Ta_k y Tp_k , en un rango de 50 a 100 s .
- H_k , en un rango de 10 a 50 Mhz.
- Pt_k , en un rango de 200 a 1000 μw .
- K_0 , en un rango de 200 a 1000 mw .
- X_k y Y_k , las coordenadas del sensor, en los rangos $[Xmin - Xmax]$ y $[Ymin - Ymax]$.

Los siguientes valores permanecen constantes durante la ejecución del algoritmo por su definición previa:

$L = 0.125$ m (velocidad de la luz /frecuencia de Transmisión).

$\beta = 1.5$

$p = 2.5$

$\lambda = 0.9$

Representación de los Cromosomas

Cada individuo de la población genética contiene una secuencia de nodos que forman el camino inter-clúster e intra-clúster, representado como una matriz de longitud fija $n \times (s + 1)$, donde cada fila representa un clúster con su secuencia de sensores descritos en las columnas de dicha fila. Por ejemplo:

114367528

083576214

263742851

Cada clúster se encuentra identificado por sus nodos cabeceras. En la columna 0 de la matriz, están representados los nodos cabecera y en las columnas restantes se encuentra la ruta interna de cada clúster. En el ejemplo, el agente visitara primero el clúster 1, luego el clúster 0 y finalmente el clúster 2. Particularmente, dentro del clúster 0 la secuencia 8-3-5-7-6-2-1-4 representa el orden en que el agente atraviesa el clúster volviendo al nodo cabecera. Aun cuando el identificador para el nodo cabecera puede repetirse en la ruta interna del clúster – en el ejemplo para la primera fila, el nodo cabecera está identificado con 1 al igual que el primer sensor del camino intra-clúster – no debe confundirse pues la primera ocurrencia se refiere al nodo cabecera y la segunda ocurrencia a un sensor interno de dicho clúster.

Mecanismo para evaluar las Funciones Objetivos a optimizar

Las funciones objetivos (1) y (2) se calculan para el camino inter-cluster e intra-cluster de un individuo de forma independiente. En consecuencia, se suman ambos valores y ese será el resultado definitivo de la función objetivo para dicho individuo.

En el caso de la función objetivo (3) no se utiliza la suma ya que la energía representativa de cada nodo depende de sus vecinos.

Descripción del Algoritmo

Siguiendo el esquema del SPEA (Zitzler and Thiele, 1998), el algoritmo mantiene una población evolutiva P y un conjunto externo de soluciones no dominadas P_{nd} halladas hasta un número de generaciones dado. Partiendo de una

población inicial aleatoria, los individuos evolucionan hasta lograr soluciones óptimas, las cuales son guardadas en Pnd . El Algoritmo 3 muestra el pseudocódigo del algoritmo propuesto. En este proceso evolutivo se van calculando los valores de las funciones objetivos para todos los individuos de la población, y utilizando los criterios de *no-dominación* mencionados anteriormente, se seleccionan los individuos candidatos a formar parte del Frente de Pareto. Debido a que las poblaciones P y Pnd son conjuntos que van evolucionando a lo largo de las generaciones, estas aparecen en función de t , el contador de generaciones.

Inicio

Leer Parametros;

$t = 0$;

Construir tabla de sensores;

Inicializar $P(t)$ y $Pnd(t)$;

Repita {

Evaluación Individuos en $P(t)$;

Marcar Individuos no dominados en $P(t)$;

Actualizar Conjunto de Pareto $Pnd(t)$;

Calcular adaptación en $P(t)$;

Calcular Fuerza en $Pnd(t)$;

Aplicar Operador de Selección por Torneo Binario en $P(t) \cup Pnd(t)$;

Aplicar Operador de Cruce en $P(t+1)$;

$P(t+1) =$ Aplicar Operador de Mutación en $P(t+1)$;

$t = t + 1$;

} Hasta el máximo de generaciones Gen ;

Fin

Algoritmo 3: *Pseudocódigo del Algoritmo Genético para Enrutamiento de Sensores Inalámbricos.*

La *Inicialización* de la población evolutiva P consiste en generar $|P|$ cromosomas en forma aleatoria. Con la representación propuesta, la inicialización consiste en crear matrices de enteros de dimensiones $n \times (s+1)$, donde cada nodo cabecera puede tomar un valor entre 0 y $n-1$, y los elementos restantes valores entre

1 y s . Este proceso hace validaciones a nivel del camino inter-cluster e intra-cluster de forma que los nodos dentro de un cromosoma no se repitan para producir caminos acíclicos.

Luego, al iniciar el bucle principal del algoritmo, los individuos de la población son evaluados utilizando las ecuaciones (1), (2) y (3). Posteriormente, el algoritmo debe hallar aquellas soluciones no dominadas de P candidatas a ser incluidas en el conjunto externo de soluciones no dominadas (Algoritmo 4). El procedimiento consiste en recorrer la población de individuos P y comparar cada uno de ellos con todos los demás. Si alguno domina al individuo que se está examinando, tal individuo recibe una marca de dominado. Los individuos que no recibieron tal marca al final del procedimiento conforman el conjunto de soluciones no dominadas de la población P .

```

Inicio
  Desde  $i = |P| - 1$  hasta 0 {
    Si el individuo  $i$  es no dominado {
      Desde  $j = i - 1$  hasta 0 {
        Si el individuo  $j$  domina al individuo  $i$  {
          Marcar al individuo  $i$  como dominado;
           $j = -1$ ;
        }
      }
    }
  }
Fin

```

Algoritmo 4: Pseudocódigo del procedimiento Marcar individuos no dominados en P .

Una vez halladas las soluciones no dominadas en P , cada una de estas es comparada con los individuos de Pnd , con el fin de *Actualizar el Conjunto de Pareto Pnd* . Si un individuo no dominado de P es no dominado respecto a Pnd , se inserta en este último, removiendo a todos los individuos de Pnd dominados por dicho individuo. Si el conjunto Pnd tiene un tamaño superior al tamaño máximo dado, se hace reducción del conjunto utilizando la estrategia de “clustering jerárquico” antes mencionada, que combina grupos adyacentes hasta que alcance el máximo permitido. Luego, los individuos centroides (individuo del cluster cuya distancia euclídea es mínima respecto a sus vecinos) de cada grupo pasaran a formar parte del Pnd ajustado.

El proceso para asignar el valor de adaptación a cada individuo de la población, al igual que en SPEA se realiza en dos fases:

- 1- El valor de adaptación de un individuo $j \in P$ se calcula como la suma de las fuerzas de todas las soluciones externas no dominadas $i \in Pnd$ que dominan a j . Se suma 1 a ese total para asegurar que los miembros de Pnd tengan mejor fuerza que los miembros de P .
- 2- a cada individuo de Pnd se le asigna un valor real $q_i \in [0,1)$, llamado fuerza o “*strength*” de dicha solución. q_i es proporcional al número de individuos $j \in P$ que son dominados por i dividido entre el tamaño de la población $|P|$.

Una vez calculados los valores de adaptación de cada individuo, se proceden a aplicar los operadores genéticos a la población P :

- **Operador de Selección:** Se propone el uso del torneo binario (Goldberg, 1989) como operador de selección, el cual escoge pares de individuos (con

reemplazo), de los cuales aquel individuo con menor valor de adaptación gana el torneo y es seleccionado para formar parte de la población en la siguiente generación.

- **Operador de Cruce:** sobre cada par de individuos seleccionados se aplica cruce en un punto (Goldberg, 1989) con una probabilidad P_c . En una generación, este operador se aplica separadamente para el camino inter-cluster e intra-cluster, removiendo las ocurrencias de sensores duplicados. En este caso, por contar con cromosomas de longitud fija, se conserva la primera ocurrencia y se sustituye la segunda ocurrencia por los sensores faltantes en el camino manteniendo el estricto orden en que aparecen para el segundo padre.

Ejemplo:

	1	2
Padres	081437625 125376148	053476281 178564123
Hijos	0814 76281 1253 64123	0534 37625 1785 76148
Hijos sin Repeticiones	0814 76253 1253 64178	0534 87621 1785 26143

- **Operador de Mutación:** Los genes de los nuevos individuos son mutados (intercambiados) aleatoriamente con una probabilidad P_m (Goldberg, 1989). el operador de mutación resulta al intercambiar las posiciones de dos sensores elegidos al azar en el camino. Por ejemplo la mutación de la secuencia 1-3-7-4-8-6-2-5 podría ser 1-3-2-4-8-6-7-5 (intercambio de 7 con 2). El operador de mutación también es aplicado independientemente para el camino intra-cluster y el inter-cluster.

Durante cada iteración, los operadores de cruce y mutación son aplicados a un camino intra-cluster o a un camino inter-cluster, con igual probabilidad. Por último, el algoritmo se detiene cuando se alcanza un número de generaciones máximo *Gen*.

CAPÍTULO V

EXPERIMENTOS Y RESULTADOS

Basado en los parámetros de entrada de Rajagopalan et al. (2005), se realiza la simulación con un tamaño para el agente móvil $M = 400$ bits, tamaño promedio de data $K = 100$ bits, ganancia en la transmisión $Gt = 2$, ganancia en la recepción $Gr = 2$, ancho de banda $B = 2$ Mbps, para un total de $N = 5$ clústeres. Para cada clúster se consideran 20 sensores internos y 2 objetivos respectivamente. Las características restantes son asignadas aleatoriamente según el mecanismo de simulación mostrado en el Capitulo IV.

Pequeñas variaciones en las probabilidades de cruce y mutación no muestran un impacto significativo en el desempeño del algoritmo. Por ello se eligen probabilidades de cruce y mutación $P_c = 0.6$ y $P_m = 0.1$.

Los resultados extraídos para 100 experimentos de 100 generaciones cada uno, con una población de 100 individuos y un tamaño máximo del frente de Pareto $Par = 10$, muestran las siguientes rutas:

Ind.	Ruta
1	2->11-36-13-18-19-8-12-20-10-9-15-14-4-7-2-5-17-16 3->17-13-5-20-14-9-16-17-8-4-23-19-11-15-10-18-12-6 0->19-8-6-1-3-17-20-15-13-14-5-11-10-2-12-7-9-18-16-4 4->2-3-7-16-12-19-8-6-20-18-13-1-17-11-4-14-9-10-15-5 1->2-14-18-11-6-13-20-8-19-3-15-9-5-16-1-17-10-4-12-7
2	2->11-36-13-18-19-8-12-20-10-9-15-14-4-7-2-5-17-16 4->3-2-7-1-12-19-8-6-20-18-13-16-17-11-4-14-9-10-15-5 3->17-13-5-20-14-9-16-17-8-4-23-19-11-15-10-18-12-6 0->19-8-6-1-3-17-20-15-13-14-5-11-10-2-12-7-9-18-16-4 1->2-14-18-11-6-13-20-8-19-3-15-9-5-16-1-17-10-4-12-7
3	2->11-36-13-18-19-8-12-20-10-9-15-14-4-7-2-5-17-16 3->17-13-5-20-14-9-16-17-8-4-23-19-11-15-10-18-12-6 0->19-8-6-1-3-17-20-15-13-14-5-11-10-2-12-7-9-18-16-4 1->2-14-18-11-6-13-20-8-19-3-15-9-5-16-1-17-10-3-12-7 4->3-2-7-16-12-19-8-6-20-18-13-1-17-11-4-14-9-10-15-5
4	2->11-36-13-18-19-8-12-20-10-9-15-14-4-7-2-5-17-16 4->3-2-7-1-12-19-8-6-20-18-13-16-17-11-4-14-9-10-15-5 0->19-8-6-1-3-17-20-15-13-14-5-11-10-2-12-7-9-18-16-4 1->2-14-18-11-6-13-20-8-19-3-15-9-5-16-1-17-10-4-12-7 3->17-13-5-20-14-9-16-17-8-4-23-19-11-15-10-18-12-6
5	2->11-36-13-18-19-8-12-20-10-9-15-14-4-7-2-5-17-16 4->3-2-7-16-12-19-8-6-20-18-13-1-17-11-4-14-9-10-15-5 0->19-8-6-1-3-17-20-15-13-14-5-11-10-2-12-7-9-18-16-4 1->2-14-18-11-6-13-20-8-19-3-15-9-5-16-1-17-10-4-12-7 3->17-13-5-20-14-9-16-17-8-4-23-19-11-15-10-18-12-6
6	2->11-36-13-18-19-8-12-20-10-9-15-14-4-7-2-5-17-16 4->3-2-7-16-12-19-8-6-20-18-13-1-17-11-4-14-9-10-15-5 0->19-8-6-1-3-17-20-15-13-14-5-11-10-2-12-7-9-18-16-4 3->17-13-5-20-14-9-16-17-8-4-23-19-11-15-10-18-12-6 1->2-14-18-11-6-13-20-8-19-3-15-9-5-16-1-17-10-4-12-7
7	4->2-3-7-16-12-19-8-6-20-18-13-1-17-11-4-14-9-10-15-5 2->11-36-13-18-19-8-12-20-10-9-15-14-4-7-2-5-17-16 0->19-8-6-1-3-17-20-15-13-14-5-11-10-2-12-7-9-18-16-4 3->17-13-5-20-14-9-16-17-8-4-23-19-11-15-10-18-12-6 1->2-14-18-11-6-13-20-8-19-3-15-9-5-16-1-17-10-4-12-7
8	2->11-36-13-18-19-8-12-20-10-9-15-14-4-7-2-5-17-16 4->3-2-7-16-12-19-8-6-20-18-13-1-17-11-4-14-9-10-15-5 0->19-8-6-1-3-17-20-15-13-14-5-11-10-2-12-7-9-18-16-4 3->17-13-5-20-14-9-16-17-8-4-23-19-11-15-10-18-12-6 1->2-14-18-11-6-13-20-8-19-3-15-9-5-16-1-17-10-4-12-7
9	4->2-3-7-16-12-19-8-6-20-18-13-1-17-11-4-14-9-10-15-5 2->11-36-13-18-19-8-12-20-10-9-15-14-4-7-2-5-17-16 0->19-8-6-1-3-17-20-15-13-14-5-11-10-2-12-7-9-18-16-4 3->17-13-5-20-14-9-16-17-8-4-23-19-11-15-10-18-12-6 1->2-14-18-11-6-13-20-8-19-3-15-9-5-16-1-17-10-4-12-7
10	2->11-36-13-18-19-8-12-20-10-9-15-14-4-7-2-5-17-16 4->3-2-7-16-12-19-8-6-20-18-13-1-17-11-4-14-9-10-15-5 1->2-14-18-11-6-13-20-8-19-3-15-9-5-16-1-17-10-3-12-7 0->19-8-6-1-3-17-20-15-13-14-5-11-10-2-12-7-9-18-16-4 3->17-13-5-20-14-9-16-17-8-4-23-19-11-15-10-18-12-6

Figura 3: Resultados del Frente de Pareto ($Par = 10$).

La evolución de los individuos del Frente de Pareto hacia las rutas óptimas se muestra en las graficas de las Figuras 4,5 y 6. Ellas muestran la convergencia individual de cada función objetivo, aún cuando son evaluadas simultáneamente por el algoritmo. Se observa una mejora paulatina a medida que avanzan las generaciones debido a que, para cada función objetivo, se obtienen valores que tienden a estabilizarse.

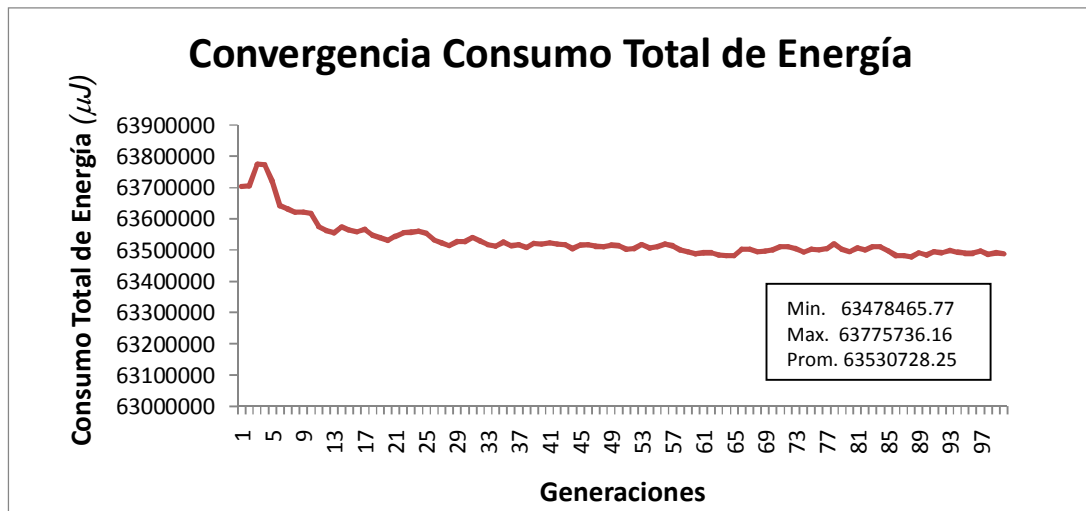


Figura 4: Minimización del Consumo total de Energía. 100 Experimentos.

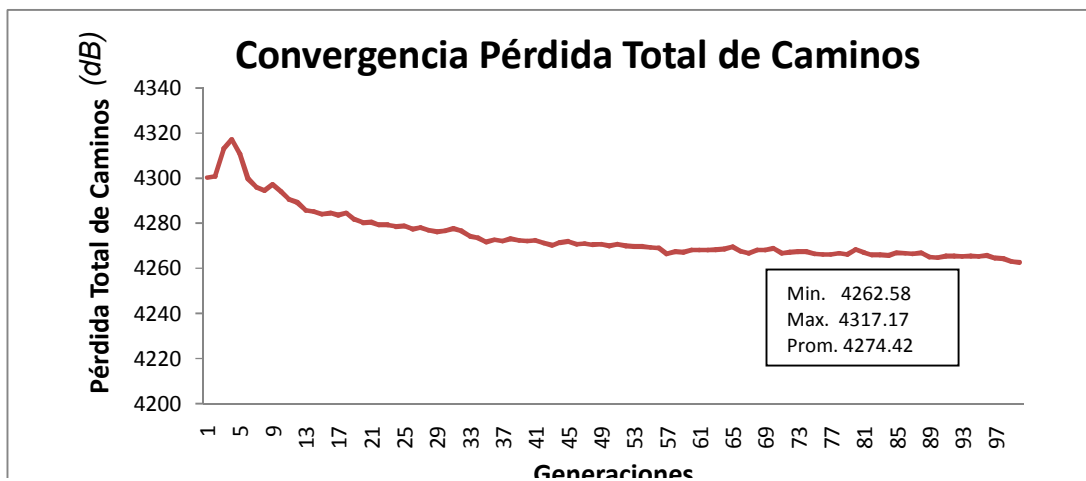


Figura 5: Minimización de la Pérdida Total de Caminos. 100 Experimentos.

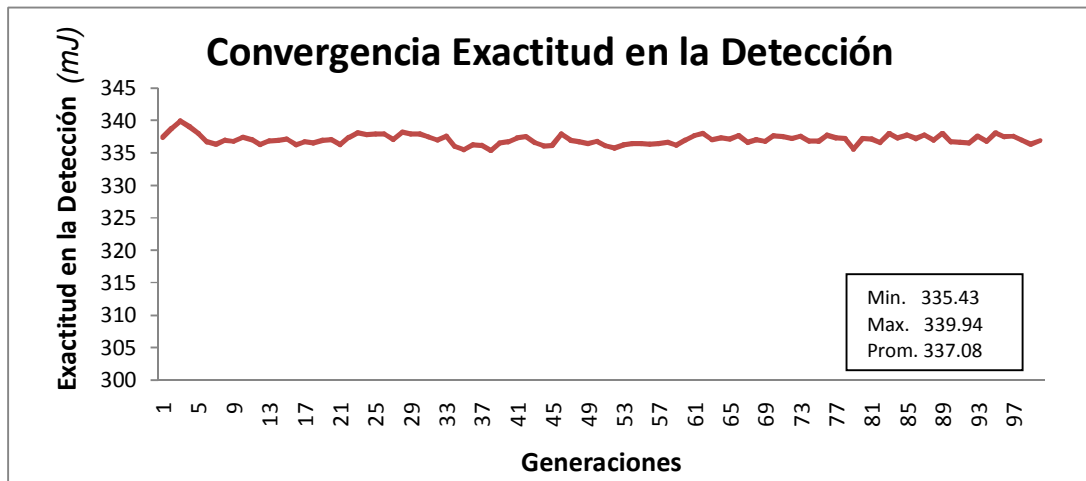


Figura 6: Maximización Exactitud en la Detección. 100 Experimentos.

En una nueva ejecución, si se incrementa el número de generaciones y el tamaño del frente de Pareto ($Gen = 800$, $Par = 50$) la tendencia se mantiene, como puede observarse en las Figuras 7, 8 y 9. Cada punto de la gráfica representa un promedio por generación, sin embargo los valores están más o menos distribuidos alrededor del promedio según muestra su desviación estándar (Figura 10). La figura 9 indica el grado de compromiso que existe entre la maximización de la exactitud en la detección y el resto de funciones objetivos, ya que al intentar mejorar afecta significativamente el resultado de las otras funciones, por eso su comportamiento zigzagueante.

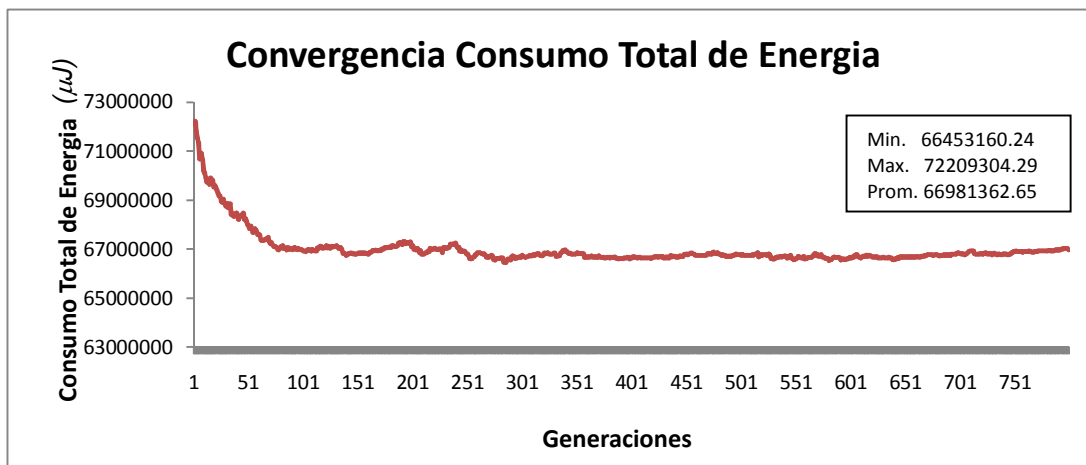


Figura 7: Minimización del Consumo total de Energía. 800 Generaciones.

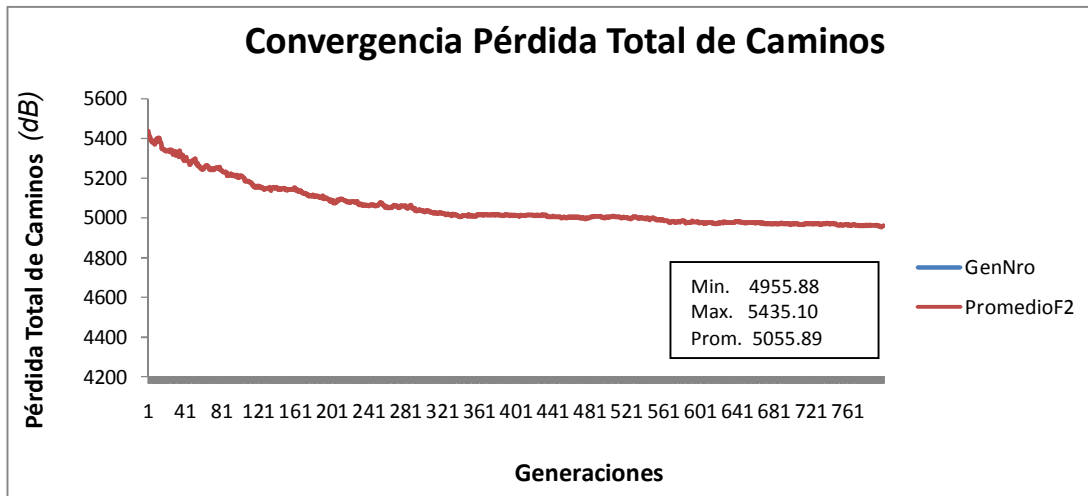


Figura 8: Minimización de la Pérdida Total de Caminos. 800 Generaciones.

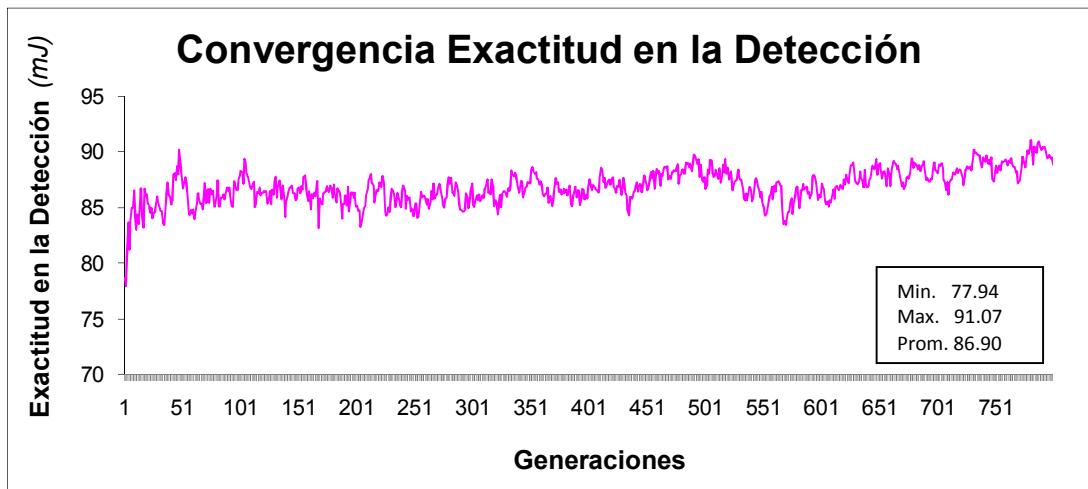


Figura 9: Maximización Exactitud en la Detección. 800 Generaciones.

Prom. Funcion/Gen	1	800	Desv Est/Gen	1	800
(1)	7.21E+07	6.70E+07	(1)	1.11E+06	8.43E+05
(2)	5435.098	4961.951	(2)	44.29604	43.78331
(3)	78.7491	88.85741	(3)	9.54699	10.88996

Figura 10: Promedio Funciones Objetivos y Desviación Estándar.

En las Figuras 11 y 12 se observan las soluciones obtenidas al graficar los individuos del conjunto no dominado luego de 800 generaciones. La Figura 11 refleja una relación cartesiana entre la Pérdida Total de Caminos y la señal de energía detectada; se reconocen soluciones tales que cuando se aumentan los caminos perdidos disminuye la exactitud en la detección y viceversa. A su vez, la Figura 12 muestra la relación entre el consumo total de energía y la Pérdida Total de Caminos, donde se observa la dependencia directamente proporcional entre ambas. Para las tres funciones objetivos consideradas, estas graficas indican que el algoritmo es capaz de encontrar rutas con alta señal de energía detectada, poco consumo de energía y poca pérdida de caminos.

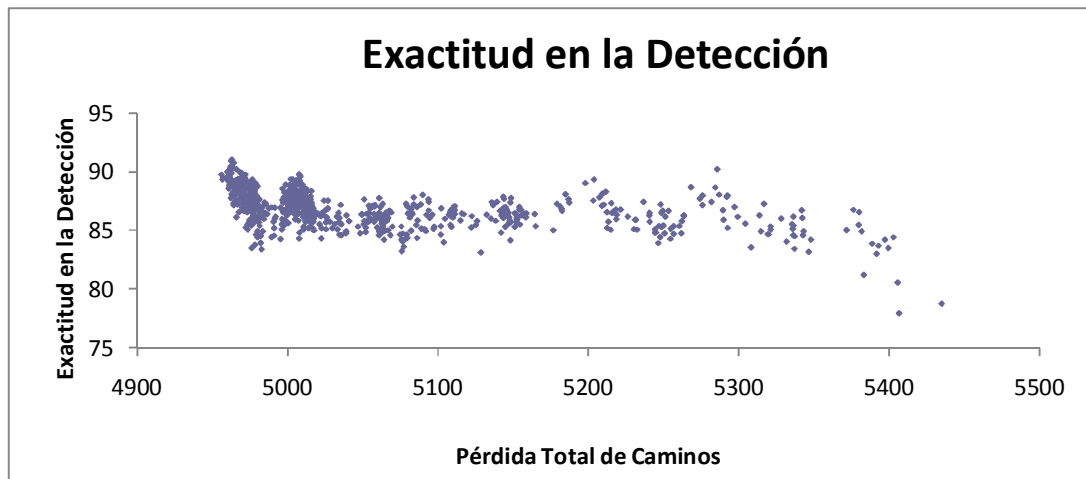


Figura 11: *Relación Cartesiana entre la Pérdida Total de Caminos y Exactitud en la Detección. Par = 50 soluciones no dominadas.*

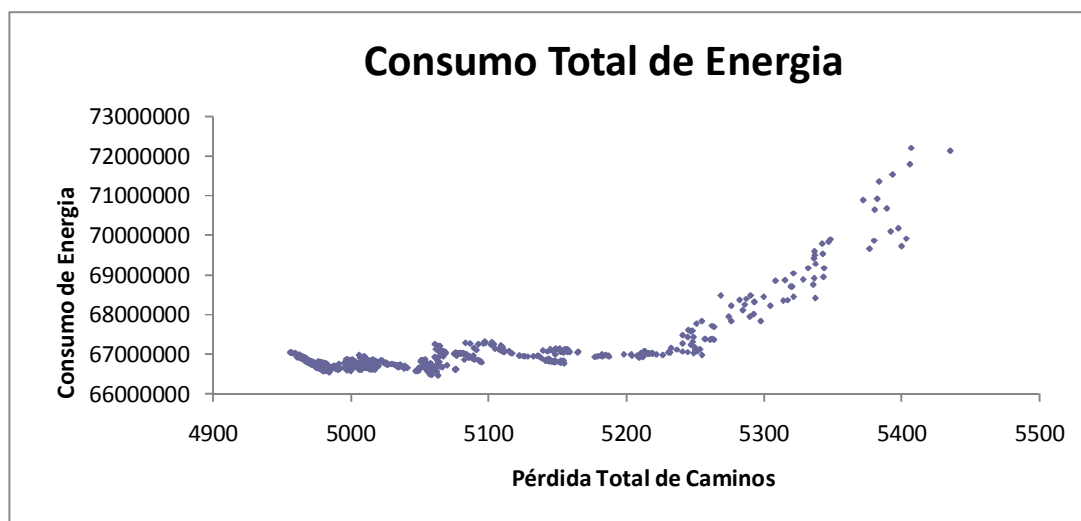


Figura 12: *Relación Cartesiana entre la Pérdida Total de Caminos y el Consumo de Energía. Par = 50 soluciones no dominadas.*

Por otro lado, se observa un adecuado espaciamiento del espacio objetivo para las soluciones no dominadas obtenidas. Esto es confirmado por el cálculo de la métrica S, utilizando valores normalizados para las funciones objetivos. La Figura 13 realiza una comparación entre la métrica S del algoritmo desarrollado y los implementados por Rajagopalan et. al (2005), para EMOCA y NSGA-II. Para la comparación se calcula el “Spacing” promedio de 800 generaciones.

nro. Objetivos	nro. Cluster	sensores p/cluster	S(EMOCA)	S(NSGA-II)	S(AP)
1	5	20	0.005	0.26	0.015
2	10	20	0.012	0.16	0.013
2	10	30	0.011	0.14	0.009
3	10	40	0.008	0.09	0.009

Figura 13: *Comparación Métrica “Spacing” entre EMOCA, NSGA-II y el Algoritmo Propuesto (AP). 800 Generaciones. (Ragajopalan et al., 2005).*

Según Rajagopalan et al. (2005) “...un algoritmo A que produce un conjunto de soluciones no dominadas ‘A’ es considerado mejor que B que produce un conjunto ‘B’ si se toman en cuenta métricas de cobertura (C), dominación (Dom) y espaciado (S)... según las cuales se considera A mejor que B si y solo $C(A,B)$ es alta, $C(B,A)$ es baja, $Dom(A,B)$ es baja y $S(A) < S(B)$...”. Los resultados muestran que el algoritmo propuesto es comparable al EMOCA en términos de su distribución en el espacio de soluciones.

Cada experimento para 100 generaciones tarda en ser ejecutado 60 minutos en promedio. Esto se debe principalmente a la reducción del frente de Pareto que se realiza en cada generación. El conjunto total de 100 experimentos dura en ejecución 4 días aproximadamente, dependiendo de los parámetros de configuración de la red.

Conclusiones

En problemas de enrutamiento en tráfico de red, en particular redes de sensores inalámbricos existen diversas metas en conflicto cuando se trata de escoger el mejor camino para enviar datos desde un nodo fuente a un nodo destino. Para la investigación fueron considerados los objetivos de minimización del consumo total de energía, minimización del camino total perdido y maximización de la exactitud en la detección.

Los algoritmos multiobjetivo de computación evolutiva son exitosos para optimizar simultáneamente objetivos en conflicto ya que se adaptan a espacios de búsqueda de cualquier tamaño y características, y gracias a su enfoque poblacional pueden generar múltiples soluciones en una iteración.

En la investigación se desarrolló un Algoritmo Genético Multiobjetivo basado en SPEA, alcanzando los objetivos propuestos, ya que se incorporó un mecanismo de evaluación de las funciones objetivos a optimizar en el camino inter-cluster e intra-cluster. Adicionalmente se configuró el algoritmo de tal forma que cada individuo de la población genética representase un posible camino a través de la red; los operadores de selección, cruce y mutación fueron adaptados para que se mantenga el principio de caminos acíclicos en el inter e intra-cluster. Para evaluar el rendimiento del algoritmo, se simuló un ambiente de operación de la red de sensores inalámbricos con sus características individuales. El resultado obtenido fue un conjunto de soluciones no dominadas con las posibles rutas que el agente pueda seguir haciendo un menor consumo de energía, con la menor pérdida de caminos entre nodos y la mayor exactitud en la detección. Finalmente, siguiendo el enfoque a posteriori, el tomador de Decisiones (estación central) elegirá entre todas ellas, la ruta de migración que el agente tomará para la fusión de datos.

Debido a la tendencia a la convergencia de las funciones objetivos consideradas, un aporte fundamental de este trabajo es que evidenció la factibilidad de la utilización del SPEA en trabajos con características similares, donde se consideren redes inalámbricas “ad hoc” y fusión de datos en el sitio local.

La efectividad demostrada por el algoritmo para encontrar rutas muy cercanas a las rutas óptimas, considerando el conflicto de las metas propuestas y la capacidad limitada de los dispositivos de la red, comprueba la utilidad de este algoritmo de enrutamiento sobre cualquier tipo y naturaleza de redes, en particular puede servir como heurística de los protocolos de la capa de red para determinar caminos en forma dinámica.

Como el enfoque utilizado es un enfoque a posteriori, el interés se centra en alcanzar un conjunto de soluciones que se extiendan por la región Pareto óptima entera. La Métrica S, mostrada en la Figura 13, indica un buen espaciamiento de las

soluciones en el espacio de búsqueda comparable a las encontradas por Rajagopalan et al. (2005). Quedaría a criterio de un dispositivo central decidir la ruta a seguir, bajo las condiciones consideradas.

El costo computacional de reducir el frente de Pareto en cada generación es alto, por lo que se propone hacer una modificación al SPEA de manera que se reduzca el frente de Pareto al final de cada experimento, formando los clústeres y extrayendo individuos representativos de acuerdo a la estrategia de Clustering Jerárquico.

Recomendaciones y Trabajos Futuros

Para mejorar la Exactitud en la detección dentro de un cluster, se propone utilizar la mediana como medida de tendencia central, entre los valores con mayor frecuencia dentro del conjunto de datos. Siendo el elemento central, el resultado es mas preciso, ya que se ignora el ruido introducido por los sensores defectuosos.

En la representación de los cromosomas, se considera la posibilidad de que estos tengan longitud variable. Al remover las ocurrencias de sensores duplicados, el cruce y la mutación se aplican a cromosomas de diferente longitud generando rutas más realistas de cara con los objetivos a optimizar.

En el diseño del ambiente de simulación, se podría imprimir mayor dinamismo a la estructura de la red. Esto se lograría inicializando medidas particulares a cada sensor de forma aleatoria, por ejemplo, el tamaño de la data recolectada, la ganancia en la transmisión y la ganancia en la Recepción.

Evaluando la habilidad del algoritmo en la exploración del espacio de soluciones, habría que estudiar qué tanta diversidad se obtiene en la población genética al aplicar cruce y mutación en el camino inter-cluster e intra-cluster de forma simultánea.

Se propone la comparación del NSGA-II, EMOCA y SPEA utilizando métricas adicionales con la finalidad de realizar aportes en cuanto a su desempeño y efectividad en diversos tipos de problemas de enrutamiento de redes.

En trabajos futuros, se pueden añadir nuevas funciones objetivo que respondan a necesidades puntuales de la fusión de sensores y el funcionamiento de la red. Así mismo, se espera aplicar el algoritmo a un caso real de redes de sensores inalámbricos heterogéneos.

GLOSARIO DE TÉRMINOS TÉCNICOS

Agente Móvil: Es un tipo especial de software que se puede ejecutar autónomamente. Una vez disparado en la red, puede migrar nodo a nodo procesando data.

Agente Móvil para Redes de Sensores Distribuidos (MADSNs): Agente Móvil que viaja a través de una red de sensores fusionando la data recolectada por cada uno de ellos.

Cabecera de Clúster: Sensor con características especiales dentro de su cluster: mayor capacidad de procesamiento, mayor nivel carga de batería en comparación con sus vecinos.

Camino Inter-Clúster: Secuencia no cíclica de cabeceras de clúster, es decir un camino donde se consideran todos los nodos sin repeticiones.

Camino Intra-Clúster: Secuencia no cíclica de sensores dentro de un clúster.

Cruce: Operador genético en el cual dos o mas soluciones padres son usadas para crear, mediante recombinación, una o mas soluciones hijas.

Datagrama: Fragmento de paquete enviado con suficiente información para que la red pueda encaminar el fragmento hacia el Equipo Terminal de Datos (ETD) receptor, de manera independiente a los fragmentos restantes. Esto no garantiza que los paquetes lleguen en el orden adecuado o que todos lleguen al destino.

Domótica: Conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación,

cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar.

Elitismo: Operador que mantenga las mejores soluciones encontradas en la población de manera que no puedan ser eliminadas en el futuro.

Fitness: Función derivada de las funciones objetivo, restricciones y otras descripciones del problema, la cual es usada en la operación de selección (reproducción) de un AE. Una solución es mejor que otra si su valor fitness es mejor. También es llamada función o valor de adaptación.

Generación: Iteración de un AE.

Hijos: Nuevas soluciones (o vectores de variables de decisión) creadas por la combinación de los operadores de cruce y mutación.

Individuo: Miembro de la población dentro del AE. Representa una posible solución al problema.

Inmótica: Incorporar a un establecimiento o edificio (ya sea hotelero, industrial, educativo o empresarial), un sistema de gestión automatizado, y desde un mismo dispositivo gestionar por ejemplo la climatización, la seguridad, la iluminación para reducir el consumo de energía, y aumentar el confort y la seguridad de los mismos.

Itinerario: La ruta de migración dentro de una red. Puede ser fija o determinada dinámicamente de acuerdo al estado actual de la red.

Mutación: Operador genético aplicado a un individuo para crear nuevas soluciones.

Nicho: Operador por el cual la presión selectiva de los miembros de la población son controlados de modo que no se permita que una única solución tome control de toda la población. Así, la técnica basada en nichos mantiene la diversidad poblacional, muy necesaria en problemas de optimización multimodales.

Padre: Solución usada durante la operación de cruce para crear soluciones hijas.

Población: Conjunto de soluciones usadas en una generación. El número de soluciones en una población es llamado *tamaño de la población*.

Pool de Apareo: Población intermedia (creada por el operador de selección), usada para crear nuevas soluciones por cruce y mutación.

Presión Selectiva: Énfasis dado a las soluciones con fitness por encima del promedio en la operación de selección. No existe una definición cuantificable, pero se considera como el número de hijos añadidos por el operador de selección a la población de mejores soluciones (frente de Pareto).

Probabilidad de Cruce: Probabilidad de ejecutar una operación de cruce. Esto significa, en promedio, la proporción de miembros de la población que participan en la operación de cruce en una generación.

Probabilidad de Mutación: Probabilidad de ejecutar una operación de mutación. Representa en promedio, la proporción de variables de decisión que participan en la operación de mutación para una solución.

Red de Sensores Heterogéneos: Red en la que cada nodo posee diferentes prestaciones en cuanto a nivel de carga de batería, capacidad de adquisición y procesamiento de datos, distancia del objetivo del cual capturan información.

Reproducción: Operador genético que representa el principio de supervivencia del más apto propuesto por Darwin, donde se eligen las mejores soluciones de la población.

Selección: Igual a Reproducción.

Sensor: Es un dispositivo que mide una cantidad física y la convierte en una señal que puede ser leída por un observador o un instrumento.

Solución: Igual a Individuo.

REFERENCIAS BIBLIOGRAFICAS

- Barán B., Duarte S., and Benítez D. (2003). *Telecommunication Network Design with Parallel Multiobjective Evolutionary Algorithm*, IFIP/ACM Latin American Networking Conference. Bolivia.
- Barán B., Vallejos J., Ramos R., and Fernández U. (2001). *Multiobjective Reactive Power Compensation*. In Proceedings of the IEEE Transmission and Distribution Conference and Exposition. USA.
- Chan H. and Perrig A. (2004). *ACE: An emergent algorithm for highly uniform cluster formation*. European workshop on sensor networks, pp. 154-171.
- Clouqueur T., Ramanathan P., Saluja K.K and Wang K.C. (2001). *Value fusion versus decision fusion for fault tolerance in collaborative target detection in sensor networks*. In Proceedings Fusion 2001 conference.
- Coello C., Dhaenens C., and Jourdan L. (2010). *Advances in Multi-Objective Nature Inspired Computing*. Studies in Computational Intelligence, Volume 272. Springer-Verlag Berlin Heidelberg.
- Coello C. (2000). *EMOO Web Page, A complete list in alphabetical order*. Evolutionary Optimization: an International Journal on Internet, 2000. Disponible en <http://www.jeo.org/emo>.
- Cormen T., Leiserson C., Rivest R. and Stein C. (2001). *Introduction to Algorithms*. Massachusetts Institute of Technology. Mc. Graw Hill. 2nd Edition.
- Darwin C. (1985). *On the Origin of Species by Means of Natural Selection*. Penguin Classics. 6° Edition.
- Deb Kalyanmoy (1999). *Evolutionary algorithms for multi-criterion optimization in engineering design*. Proceedings of the Evolutionary Algorithms in Engineering and Computer Science (EUROGEN'99).
- Deb Kalyanmoy (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley and Sons.
- Deb K., Branke J., Miettinen K. and Słowiński R. (2008). *Multiobjective Optimization Interactive and Evolutionary Approaches*. Springer Berlin Heidelberg. New York.

- Dijkstra E. (1959). *A note on two problems in connexion with graphs*. Numerische Mathematik, Vol. 1, pág. 269-271.
- Friss H.T. (1946). *A note on a simple transmission formula*. In Proceedings of IRE.
- Fonseca C., and Fleming P. (1995). *Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms I: A Unified Formulation*. Technical Report 564, Department of Automatic Control and System Engineering, University of Sheffield. Reino Unido.
- Fonseca C., and Fleming P. (1995). *An Overview of Evolutionary Algorithm in Multiobjective Optimization*. Evolutionary Computation, Vol. 3, N° 1, pág. 1-16.
- Forouzan, Behrouz (2001). *Transmisión de Datos y Redes de Computadoras*. Mc. Graw Hill. 2da. Edición Cap 21. España.
- Gimeno José Manuel. (2010). *La Inteligencia Artificial toma el Control*. Disponible en: <http://www.laflecha.net/articulos/ciencia/IA/> . Fecha de visita: 03/05/2010.
- Goldberg D. (1989). *Genetic Algorithm is Search, Optimization & Machine Learning*. Addison Wesley.
- Haupt R. and Haupt S. (1998). *Practical Genetic Algorithms*. John Wiley & Sons, Inc.
- Heitkoetter J. and Heitkoetter B. (2000). *The Hitch-Hiker's Guide to Evolutionary Computation: A List of Frequently Asked Questions (FAQ)*. Disponible en: <http://www.cs.cmu.edu/Groups/AI/html/faqs/ai/genetic/part2/faq.html>, Fecha de Visita: 31/05/2010.
- Holland J. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press Edition.
- Horn J. (1997). *The nature of niching: genetic algorithms and the evolution of optimal, cooperative population*. Ph.D Thesis, University of Illinois at Urbain Champaign, USA.
- Morse J. (1980). *Reducing the size of the non-dominated set: pruning by clustering*. Computer and Operation Research 7(1-2), 55-66.
- NASA. (2004) *Project X-45 Unmanned Combat Air Vehicle (UCAV)*. Disponible en: <http://www.nasa.gov/centers/dryden/research/X45A/index.html>. Fecha de visita: 03/05/2010.

- Nirupama B. and Sanjay J. (2005). *Wireless Sensor Networks: A Systems Perspective*. Artech House.
- Pareto, V. (1896). *Cours D'Economie Politique*. vol. I, II. F. Rouge, Lausanne.
- Qi H., S.S. Iyengar, and K. Chakrabarty (2001). *Multiresolution data integration using mobile agents in distributed sensor networks*. In IEEE Trans. Systems, Man and Cybernetics Part C: Applications and Rev. vol. 31, No. 3.
- 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.
- Rappaport T. S. (1996). *Wireless Communications Principles and Practice*. 1st. Edition, Prentice Hall. New Jersey.
- Sotelo A., Barán B. and Von Lucken C. (2002). *Multiobjective Evolutionary Algorithms for Pump Scheduling Optimization*. 3rd International Conference on Engineering Computational Technology ECT-2002. Rca. Checa.
- 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.
- Wu Q., N.S.V Rao, J. Barhen, S.S. Iyengar, V.K. Vaishnavi, H. Qi and K. Chakrabarty. (2004). *On computing mobile agent routes for data fusion in distributed sensor networks*. IEEE Trans. Knowledge and Data Engineering, vol. 16, no. 6, pp. 740-753.
- Zitzler E. and Thiele L. (1998). *An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach*. TIK Report. Nro. 43.