

**METAHEURISTICA BASADA EN AUTOMATAS FINITOS Y ALGORITMOS
GENETICOS PARA LA OPTIMIZACION TRI-OBJETIVO DEL PROBLEMA DEL
AGENTE VIAJERO SIMETRICO MULTIOBJETIVO**

ANANGÉLICA ISABEL CHINCHILLA CAMARGO

**UNIVERSIDAD DEL NORTE
DIVISIÓN DE INGENIERÍAS
MAESTRÍA EN INGENIERIA INDUSTRIAL
BARRANQUILLA
2011**

**METAHEURISTICA BASADA EN AUTOMATAS FINITOS Y ALGORITMOS
GENETICOS PARA LA OPTIMIZACION TRI-OBJETIVO DEL PROBLEMA DEL
AGENTE VIAJERO SIMETRICO MULTIOBJETIVO**

ANANGÉLICA ISABEL CHINCHILLA CAMARGO

TESIS DE GRADO

*Presentado como requisito de grado para optar al título de Magister en Ingeniería
Industrial*

Director

MSc Ing. Elías David Niño Ruiz

**UNIVERSIDAD DEL NORTE
DIVISIÓN DE INGENIERÍAS
MAESTRÍA EN INGENIERIA INDUSTRIAL
BARRANQUILLA**

2011

Nota de aceptación

Elías David Niño Ruiz, MSc
Director del proyecto

Ph.D Carmenza Luna Amaya
Coordinador Programa de Maestría en
Ingeniería Industrial

Corrector

Jurado

Jurado

20 de Mayo de 2011

AGRADECIMIENTOS

Agradezco a Dios por mostrarme el camino correcto a lo largo de mi vida y permitirme aprender a levantarme de cada error cometido y de cada batalla perdida.

A mi familia por ser el árbol en el que puedo apoyarme cada día de mi vida. A la familia de Javier por ser parte mi familia desde el momento en que entré en sus vidas.

A Javier por estar en cada momento sin preguntar razones, ni juzgar de manera negativa mis acciones y en general, por querernos tanto.

Gracias a mis amigos por ayudarme a andar por el difícil camino de la vida.

No menos importante, agradezco a Elías por su constante apoyo a lo largo de los años.

RESUMEN

En la actualidad solo existe una metaheurística basada en Autómata Finito Determinista que permite modelar y describir un espacio de soluciones factibles para optimizar problemas combinatorios multi-objetivo tipo hard, llamada Metaheurística de Intercambio Determinista sobre Autómatas (MIDA).

La tesis presentada a continuación se basa en MIDA tomando el espacio de soluciones factibles generado por este y realizando un cruzamiento basado en algoritmos genéticos. Se trata de un modelo para la optimización de problemas combinatorios tri-objetivo del tipo del Problema del Agente Viajero Simétrico Multi-objetivo, por medio de la implementación de una metaheurística basada en autómatas finitos y algoritmos genéticos (AGMIDA) y el análisis del comportamiento al cambiar las proporciones de pesos en las funciones objetivos del problema.

Por último se contrastan los resultados obtenidos por AGMIDA con otras técnicas relacionadas con el tema, por medio del análisis de métricas e instancias de uso internacional que permiten comparar la efectividad de los métodos, en donde la modificación que se hace a MIDA por medio de AGMIDA permite obtener mejores resultados en contraste con las otras técnicas de impacto mundial.

TABLA DE CONTENIDO

| | |
|--|----|
| 1. INTRODUCCION | 10 |
| 1.1. PLANTEAMIENTO DEL PROBLEMA | 10 |
| 1.2. OBJETIVOS | 11 |
| 1.2.1. Objetivo General | 11 |
| 1.2.2. Objetivos Específicos..... | 11 |
| 1.3. CONTRIBUCIONES | 12 |
| 1.4. ANTECEDENTES | 12 |
| | |
| 2. MARCO TEÓRICO | 22 |
| 2.1. OPTIMIZACIÓN | 22 |
| 2.1.1. Optimización Mono-Objetivo..... | 24 |
| 2.1.2. Optimización Multi-Objetivo | 26 |
| 2.1.2.1. Frente de Pareto..... | 27 |
| 2.1.2.1.1. Métricas..... | 30 |
| 2.1.2.1.1.1. Generación de Vectores No Dominados | 30 |
| 2.1.2.1.1.2. Distancia Generacional | 30 |
| 2.1.2.1.1.3. Distancia Generacional Inversa | 31 |
| 2.1.2.1.1.4. Espaciamiento | 31 |
| 2.1.2.2. Técnicas Clásicas..... | 32 |

| | |
|--|----|
| 2.1.2.2.1. – Restricción | 32 |
| 2.1.2.2.2. Suma con Pesos | 33 |
| 2.2. OPTIMIZACIÓN COMBINATORIA..... | 34 |
| 2.2.1. Técnicas Heurísticas | 36 |
| 2.2.2. Técnicas Metaheurísticas | 37 |
| 2.2.2.1. Algoritmos Genéticos..... | 37 |
| 2.2.2.2. Recocido Simulado | 40 |
| 2.2.2.3. Colonia de Hormigas..... | 43 |
| 2.2.2.4. Búsqueda Tabú | 48 |
| 2.2.2.5 Greedy Randomized Adaptive Search Procedure..... | 51 |
| 2.3. AUTÓMATAS..... | 55 |
| 2.3.1. Autómata Finito Determinista | 55 |
| 2.3.1.1. Función de Transición Extendida..... | 56 |
| 2.3.1.2. Lenguaje Aceptado por un AFD | 56 |
| 2.3.1.3. Formas de Representación de los AFDs | 57 |
| 2.3.1.3.1. Tablas de Transiciones | 57 |
| 2.3.1.3.2. Diagramas de Transición | 57 |
| 2.3.2. Autómata Finito Determinista de Intercambio..... | 59 |
| 2.3.3. Autómata Finito Determinista Multi-Objetivo | 63 |
| 2.3.3.1. Conjunto de Estados | 63 |
| 2.3.3.2. Alfabeto Finito de Entrada | 65 |
| 2.3.3.3. Función de transición..... | 67 |
| 2.3.3.4. Conjunto de Estados Iniciales | 69 |

| | | |
|----------|--|----|
| 2.3.3.5. | Conjunto de Funciones Objetivos | 69 |
| 3. | METAHEURISTICA PROPUESTA..... | 72 |
| 3.1 | METAHEURISTICA BASADA EN AUTOMATAS FINITOS Y ALGORITMOS GENETICOS PARA LA OPTIMIZACION TRI-OBJETIVO DEL PROBLEMA DEL AGENTE VIAJERO SIMETRICO MULTI OBJETIVO | 72 |
| 3.2. | ANALISIS DE LA COMPLEJIDAD DE AGMIDA | 79 |
| 4. | EXPERIMENTACION Y ANALISIS DE RESULTADOS | 83 |
| 4.1. | ESPECIFICACIÓN DEL AFDM PARA LA PRUEBA | 84 |
| 4.2. | COMPARACIÓN DE LOS RESULTADOS DE AGMIDA CON LAS METAHEURÍSTICAS SMOSA, CMOSA, UMOSA Y EMOSA | 85 |
| 5. | CONCLUSIONES Y TRABAJOS FUTUROS..... | 89 |
| 6. | REFERENCIAS BIBLIOGRAFICAS..... | 91 |

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1- Búsqueda Aleatoria de Soluciones Usando CMOSAs..... | 16 |
| Figura 2- Búsqueda Aleatoria de Soluciones Usando UMOSA | 17 |
| Figura 3- Búsqueda Aleatoria de Soluciones Usando SMOSA | 17 |
| Figura 4- Conjuntos Soluciones para un Problema Mono-Objetivo | 26 |
| Figura 5- Conjunto de Soluciones Mono-Objetivo..... | 28 |
| Figura 6. Conjunto de Soluciones Multi-Objetivo | 28 |
| Figura 7. Frente de Pareto Tri-Objetivo..... | 29 |
| Figura 8- Algoritmo Genético | 39 |
| Figura 9- Algoritmo Básico del Recocido Simulado para Minimización | 42 |
| Figura 10- Pseudocódigo del Algoritmo Original del As | 44 |
| Figura 11- Comportamiento Adaptativo de las Hormigas | 46 |
| Figura 12- Algoritmo de Búsqueda en el Entorno | 50 |
| Figura 13- Algoritmo de Búsqueda Tabú..... | 50 |
| Figura 14- Diagrama de Flujo del Algoritmo GRASP..... | 52 |
| Figura 15- Estructura Básica de un Algoritmo GRASP | 54 |
| Figura 16- Diagrama de Transiciones del Ejemplo | 59 |
| Figura 17- Estructura de un estado de un AFD – I | 59 |
| Figura 18- Representación del AFDI para el vector de entrada | 62 |
| Figura 19- Estructura de un estado | 63 |
| Figura 20- Representación del Espacio de Soluciones Factibles para el Problema del Ejemplo | 64 |
| Figura 21- AFDM construido a partir del estado inicial del ejemplo 1 | 68 |
| Figura 22- Conjunto de Soluciones Factibles para el Problema del Ejemplo 1 | 71 |
| Figura 23- Cruzamiento Realizado en AGMIDA..... | 74 |

| | |
|--|----|
| Figura 24- Framework de la Metaheurística AGMIDA | 76 |
| Figura 25- Framework del método | 78 |
| Figura 26- Algoritmo del Vecino Más Cercano | 85 |
| Figura 27- Contraste Visual del FP obtenido por AGMIDA en la instancia KROABC100, contra los FPs obtenidos por los algoritmos CMOSa, SMOSA, UMOSA y EMOSA | 86 |
| Figura 28- Contraste visual del FP obtenido por AGMIDA en la instancia KROBCD100, contra los FPs obtenidos por los algoritmos CMOSa, SMOSA, UMOSA y EMOSA | 87 |

LISTA DE TABLAS

| | |
|--|----|
| Tabla 1- Taxonomía de los Algoritmos MOACOS | 15 |
| Tabla 2- Los Mejores Resultados No Dominados de MOSCA2b | 18 |
| Tabla 3- Comparación de Resultados de las Técnicas BCA, PACO, CA, BCM, MOAQ, MACS, MONA, NSGA-II, USBC, SPEA2 y MIDA Utilizando Métricas para la Instancia KROAB100 | 20 |
| Tabla 4- Comparación de Resultados de las Técnicas por BCA, PACO, CA, BCM, MOAQ, MACS, MONA, NSGA-II, USBC, SPEA2 y MIDA Utilizando Métricas para la Instancia KROAB50 | 20 |
| Tabla 5- Comparación de Resultados de las Técnicas por BCA, PACO, CA, BCM, MOAQ, MACS, MONA, NSGA-II, USBC, SPEA2 y MIDA Utilizando Métricas para la Instancia KROBC100..... | 21 |
| Tabla 6- Comparación de Promedios de los Resultados de las Técnicas por BCA, PACO, CA, BCM, MOAQ, MACS, MONA, NSGA-II, USBC, SPEA2 y MIDA con las Diferentes Métricas | 21 |
| Tabla 7- Tabla de Transiciones | 57 |
| Tabla 8- Tabla de Transiciones del Ejemplo | 58 |
| Tabla 9- Función para el AFD I con vector de entrada | 61 |
| Tabla 10- Evaluación de los vectores en la función objetivo | 62 |
| Tabla 11-Función para del AFDM del ejemplo 1 | 68 |
| Tabla 12- Función Aplicada a los Estados , , del AFDM del Ejemplo 1..... | 70 |
| Tabla 13- Variables utilizadas en AGMIDA | 79 |
| Tabla 14- Instancias de la TSPLIB utilizadas en las pruebas de AGMIDA | 84 |

| | |
|---|----|
| Tabla 15- Comparación de los resultados obtenidos por MIDA, EMOSA, CMOSA, SMOSA y UMOSA para la instancia KROABC100 utilizando las métricas GVND, ESPACIAMIENTO, GD e IGD | 86 |
| Tabla 16- Comparación de los resultados obtenidos por MIDA, EMOSA, CMOSA, SMOSA y UMOSA para la instancia KROBCD100 utilizando las métricas GVND, ESPACIAMIENTO, GD e IGD | 87 |
| Tabla 17- Promedios de los resultados obtenidos por AGMIDA, EMOSA, CMOSA, SMOSA y UMOSA para las instancias KROABC100 y KROBDC100 utilizando las métricas GVND, ESPACIAMIENTO, GD e IGD | 88 |

1. INTRODUCCION

1.1. PLANTEAMIENTO DEL PROBLEMA

Cada acción de la vida cotidiana conlleva un proceso. En consecuencia, todos los días nos enfrentamos a solución de problemas y de manera intuitiva escogemos la solución óptima: la ruta de bus que llega más rápido al destino o la ropa a usar en un día de sol. Al optimizar se puede minimizar o maximizar los recursos por ejemplo al ahorrar dinero se está minimizando gasto y al tiempo maximizando los recursos disponibles.

En la industria la optimización es utilizada en gran número de aéreas tales como el despacho económico en sistemas eléctricos¹, seleccionar una ruta óptima de recorrido de varias ciudades² o cortar piezas irregulares en la minería³. Todos estos problemas al optimizarlos lo que se busca es hallar la mejor forma de lograr uno o varios objetivos teniendo en cuenta las restricciones del sistema. Es decir, optimizar costos, tiempo, maquinaria, espacio, personal o cualquier otro recurso implicado dentro del proceso.

¹ Correa Posada, Carlos Mario. "Modelo de Optimización para las Plantas Térmicas de Generación de Ciclo Combinado en el Despacho Económico", Universidad nacional de Colombia. Medellín. 2009.

² Peiyong Chen; Xinming Xu. "A Hybrid Algorithm for Multi-Depot Vehicle Routing Problem", Service Operations and Logistics, and Informatics, IEEE/SOLI 2008, IEEE International Conference, vol. 2, pag. 2031, 25 Noviembre, 2008.

³ Lastres Aleaga, Arlys Michel; Cordovés Garcias, Alexis; Arzola Ruis, José. "Optimización de la Distribución y Corte de Piezas Irregulares en Chapas para la Industria Minera", Minería y Geología, vol.26, no.3, pag. 68-80, Julio-Septiembre, 2010.

La complejidad del problema de optimización es proporcional al número de objetivos y restricciones del proceso, por ello el espacio de soluciones factibles será cada vez más grande y encontrar al óptimo global será cada vez más difícil.

Se propone una metaheurística basada en autómatas finitos y algoritmos genéticos que permita la optimización tri-objetivo del problema del agente viajero simétrico y comparar los resultados obtenidos con trabajos especializados en el tema.

1.2. OBJETIVOS

1.2.1. Objetivo General

Diseño e implementación de una metaheurística basada en Autómatas Finitos Deterministas y Algoritmos Genéticos para la optimización en tres objetivos del problema del Agente Viajero Simétrico Multi-objetivo.

1.2.2. Objetivos Específicos

1. Diseñar un Autómata Finito Determinista que permita representar el espacio de soluciones factibles del problema del Agente Viajero Simétrico con Tres Objetivos.
2. Diseñar una metaheurística híbrida entre Metaheurística de Intercambio Determinístico y Algoritmos Genéticos para la Optimización Multiobjetivo del problema del Agente Viajero Simétrico Multiobjetivo.

3. Comparar los resultados obtenidos por metaheurísticas multiobjetivo de la literatura especializada.

1.3. CONTRIBUCIONES

Se espera que el autómata diseñado permita encontrar una gran cantidad de soluciones No Dominadas que permitan la optimización de problemas combinatorios con complejidad polinomial tipo hard. También se espera que el programa obtenido sea de gran utilidad en la industria, principalmente en el sector productivo, en donde frecuentemente encontramos problemas combinatorios tri-objetivo del tipo del problema del agente viajero simétrico multiobjetivo. Además, minimizar el consumo de recursos no renovables tales como el gas, el petróleo o el carbón, por medio de la optimización de los diferentes procesos en donde se involucran dichos recursos.

1.4. ANTECEDENTES

Los esfuerzos investigativos se han invertido en la inteligencia y métrica aplicadas a técnicas metaheurísticas. El estudio de la representación del espacio factible de soluciones de problemas combinatorios de optimización no ha sido explorado de manera exhaustiva. Uno de los estudios más reciente del tema fue hecho por Niño y Ardila⁴ en el año 2009, en donde proponen un Autómata Finito Determinista de Intercambio (AFD-I) que permite representar el espacio de soluciones factibles para un problema combinatorio, lo que ha sido de gran ayuda en la industrial para solucionar cierto tipo de dificultades. Sin embargo,

⁴ Niño, Elías; Ardila, Carlos. “Algoritmo Basado en Autómatas Finitos Deterministas para la obtención de óptimos globales en problemas de naturaleza combinatoria”, Revista de Ingeniería y Desarrollo, No 25, pag. 100 – 114. ISSN 0122 – 3461, 2009.

solo soluciona problemas combinatorios mono-objetivo, lo cual no permite resolver problemas más complejos encontrados en la industria.

Dorigo en el año 1996⁵, en su tesis doctoral, propuso un algoritmo basado en el comportamiento natural de las hormigas. Esta tesis fue nombrada algoritmo basado en Colonia de Hormigas - ACO por sus siglas en inglés. Hoy en día se pueden encontrar diferentes técnicas metaheurísticas, gracias a las investigaciones hechas por Dorigo y cuya finalidad es aproximar las soluciones a los óptimos del problema. También se encuentra una gran cantidad de algoritmos basados en este sistema, encaminados a problemas multi-objetivo (Multi – Objective Ant Colony Optimization - MOACO).

Iredi en el año 1998⁶, propone una técnica para solucionar el problema multi – objetivo del ruteo de vehículos, llamada Hormiga – Bicriterio (BCA). El BCA aplica el concepto de feromona de las hormigas, generando una matriz de ésta por cada objetivo del problema. Es decir, si el problema tiene n objetivos, se generaran n matrices de feromona. Iredi especifica una heurística propia donde este obliga a las hormigas visitar diferentes regiones del Frente de Pareto.

Mariano y Morales en el año 1999⁷ desarrollan un modelo Multi – objetivo llamado Multi – objective Ant – Q ó MOAQ, con la idea de solucionar un problema de las redes de distribución del agua. La finalidad de MOAQ, es que una vez asignadas funciones a una

⁵ Dorigo, Marco; Maniezzo, Vittorio; Colomi, Alberto. “The Ant System: Optimization by a Colony of Cooperating Agents” Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions, vol.26, no.1, pag. 29-41, Febrero, 1996.

⁶ Iredi, Steffen ; Merkle, Daniel; Middendorf, Martin. “Bi-criterion optimization with multi colony ant algorithms”, First International Conference on Evolutionary Multi-criterion Optimization, Lecture Notes in Computer Science, pag 359–372, 2001

⁷ Mariano, Carlos; Morales, Eduardo. “A multiple objective Ant-Q algorithm for the design of water distribution irrigation networks”, Technical Report HC-9904, Instituto Mexicano de Tecnología del Agua, Mexico, Junio, 1999

familia de agentes, asignarles a su vez a cada cual uno de los objetivos del problema, ver el comportamiento de las demás funciones cada vez que un agente encuentra una optimización del problema en una función.

Gambardella en el año 1999⁸, desarrolla un algoritmo con la finalidad de solucionar el problema de ruteo de vehículos llamado (Multiple ant colony system for vehicle routing problem with time Windows ó MACS-VRPTW). MACS propone una colonia n de hormigas para n funciones objetivo donde cada colonia de hormigas esta encargada de optimizar su función objetivo del problema.

Doerner en el año 2003⁹, idea una técnica de optimización de pareto, utilizando la Colonia de Hormigas, esta es llamada (Pareto – Ant Colony Optimization ó PACO). Esta técnica consta de varias matrices de feromona, una para cada objetivo, donde adicionalmente un vector de pesos mezcla proporciones de las matrices de feromona para este determinar la próxima ciudad a visitar por parte de cada hormiga a partir de una ciudad origen con la finalidad de comparar las soluciones ya que esto ayuda a explorar nuevas regiones en el Frente de Pareto.

Cardoso en el año 2003¹⁰, aplicando un algoritmo idea la optimización de redes de computación (Multi – Objective Network ACO ó MONA). Donde se puede suponer que cada hormiga es un mensaje que se transmitirá por la red y este mensaje teniendo en cuenta las matrices de feromonas se enruta de acuerdo al número de objetivos del problema.

⁸ Gambardella, Luca; Taillard, Éric; Agazzi, Giovanni. “MACS-VRPTW: A Multiple Colony System For Vehicle Routing Problems With Time Windows”, *New Ideas in Optimization*, McGraw-Hill, 1999, pag.. 73–76, 1999.

⁹ Doerner, Karl; Gutjahr, Walter; Hartl, Richard; Strauss, Christine; Stummer, Christian. “Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection”, *Annals of Operations*, pag. 79–99, 2004

¹⁰ Cardoso, Pedro; Jesus, Mario; Marquez, Alberto .“MONACO- Multi-Objective Network Optimization based on an ACO”, *Proc. X Encuentros de Geometría Computacional*, Sevilla, Spain, Junio 16–17, 2003.

Doerner en el año 2003¹¹, desarrolla la Competencia de Hormigas que es un esquema competitivo de Colonias de Hormigas llamado (COMPETants ó CA). Este fue aplicado en problemas con dos objetivos, donde por cada objetivo existe una colonia de hormigas, la colonia que obtenga las mejores soluciones en una iteración, se tendrá mayor cantidad de hormigas siguiente iteración.

Iredi¹² idea un algoritmo en las Colonias de Hormigas basado en Bicriterio (BCA) y técnicas evolutivas llamado Bicriterion MC (BCM). Esta una generalización del BCA aplicado a problemas multi – objetivo utilizando cruzamiento de soluciones para evitar óptimos locales. La tabla 1 ilustra la taxonomía de los algoritmos previamente mencionados.

| | Una sola matriz heurística | Múltiple matrices heurística |
|--|-----------------------------------|---|
| Una sola matriz de feromona. | MOACOM | MOAQ – MACS |
| Múltiples matrices de feromona. | PACO MONA | BicriterionAnt (BA) UnsortBicriterion (USBC) CompentAnts (CA) MACS – VRPTW |

Tabla 1- Taxonomía de los Algoritmos MOACOS

SPEA2¹³ y NSGA- II, son unas nuevas técnicas modernas permiten una nueva optimización de problemas con el principal objetivo de converger soluciones optimas

¹¹ Doerner, Karl; Hartl, Richard; Reimann, Marc. “Are COMPETants more competent for problem solving?— The Case of Full Truckload Transportation”, Central European Journal of Operations Research, pag. 115–141, 2003

¹² Iredi, Steffen ; Merkle, Daniel; Middendorf, Martin. “Bi-criterion optimization with multi colony ant algorithms”, First International Conference on Evolutionary Multi-criterion Optimization, Lecture Notes in Computer Science, pag 359–372, 2001.

¹³ Tse Guan Tan; Hui Keng Lau; Teo, Jason. "Cooperative Coevolution for Pareto Multiobjective Optimization: An Empirical Study Using SPEA2", TENCON 2007 - 2007 IEEE Region 10 Conference, pag.1-4, Octubre 30 -Noviembre 2, 2007.

rápidamente. Sin embargo, una rápida convergencia puede culminar con la obtención de óptimos locales y no globales.

Para la optimización de multi – objetivo actual, son usadas técnicas clásicas como las estrategias de búsqueda local. Un ejemplo de estas técnicas clásicas es la de Recocido Simulado Multi – Objetivo (Multi – objective simulated annealing ó MOSA). Los MOSA son técnicas poderosas debido a su fiabilidad ya que estas evitan en lo posible óptimos locales.

En 1998 Czyzak¹⁴ desarrolla un algoritmo de recorrido simulado al que llamo CMOSA, que es una técnica de SA y trabaja con direcciones adaptativas dependiendo de los valores de las funciones objetivos. La búsqueda de CMOSA se muestra a en la figura 1.

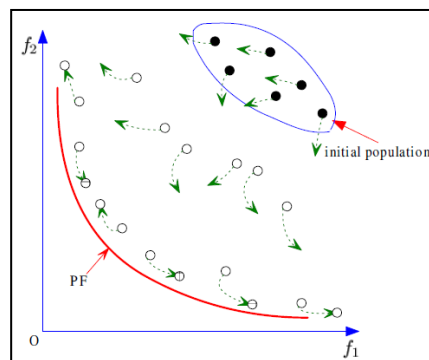


Figura 1- Búsqueda Aleatoria de Soluciones Usando CMOSA

Ulungu en el año 1999¹⁵, propone la optimización de un problema combinatorio multi – objetivo mediante la optimización de cada objetivo por medio de una corrida de un SA. Para mantener la diversidad de las soluciones no dominadas, UMOSA trabaja con un vector

¹⁴ Czyzak, Piotr; Jaskiewicz, Adrezej. “Pareto simulated annealing - a metaheuristic technique for multiobjective combinatorial optimization”, Journal of Multi-Criteria Decision Analysis, vol. 7, no. 1, pag. 34–37, Diciembre, 1998.

¹⁵ Ulungu; Teghem; Fortemps; Tuytens, “MOSA method: A tool for solving multiobjective combinatorial optimization problems,” Journal of Multi-Criteria Decision Analysis, vol. 8, no. 4, pp. 221–236, 1999.

de pesos asignado de manera balanceada a cada uno de los objetivos. La búsqueda de óptimos de UMOSA se muestra en la figura 2.

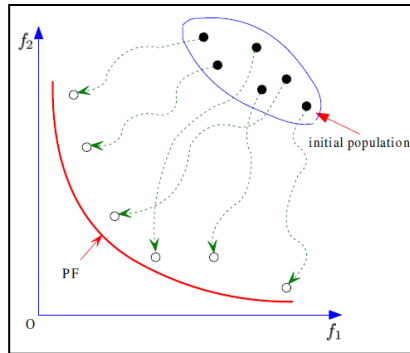


Figura 2- Búsqueda Aleatoria de Soluciones Usando UMOSA

Serafini en el año 2008¹⁶ propone una aproximación a MOSA. Esta aproximación se llama Serafinis MOSA (SMOSA) donde esta busca optimizar problemas combinatorios asignando un peso a cada una de las funciones en cada iteración. La figura 3 ilustra la dirección aleatoria de búsqueda de SMOSA.

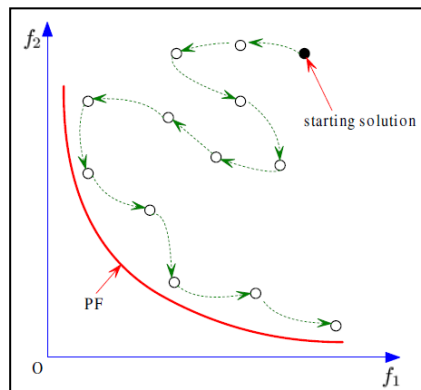


Figura 3- Búsqueda Aleatoria de Soluciones Usando SMOSA

¹⁶ Bandyopadhyay, Sanghamitra; Saha, Sriparna; Maulik, Ujiwal; Deb, Kalyanmoy. "A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA", Evolutionary Computation, IEEE Transactions, vol.12, no.3, pag.269-283, Junio, 2008

Otro aporte importante al problema del agente viajero multi-objetivo fue hecho por Borgulya en el año 2007¹⁷. También desarrolló otro algoritmo del tipo evolutivo, basado en la asignación cuadrática bi-objetivo a partir del mismo problema¹⁸, que utiliza el método de memoria colectiva o virtual perdedor, que permite trabajar con una mayor cantidad de valores discretos. El algoritmo fue llamado MOSCA2b y a continuación se muestran una comparación con el algoritmo llamado genetic local search for multiple objective combinatorial optimization (MOGLS) tomando como medida de rendimiento el indicador binario ϵ - indicador y como instancias KROAB50, KROAB100, KROAD50, KROAD100 para dos objetivos y KROABC100, KROABD100 para tres objetivos, ϵ_1 da $I_\epsilon(B, A)$, ϵ_2 da $I_\epsilon(A, B)$ donde A es el resultado de MOSCA2b y B es el resultado de MOGLS, no hace referencia al número de objetivos utilizados, los resultados son mostrados en la tabla 2.

| N° | INSTANCIA | ϵ_1 | ϵ_2 | Ndn |
|----|-----------|--------------|--------------|------|
| 2 | KROAB50 | 1.082 | 1 | 253 |
| | KROAD50 | 1.002 | 0.999 | 213 |
| | KROAB100 | 0.991 | 1.019 | 319 |
| | KROAB100 | 0.974 | 0.997 | 312 |
| 3 | KROABC100 | 0.979 | 1.025 | 2565 |
| | KROABD100 | 0.977 | 0.998 | 2515 |

Tabla 2- Los Mejores Resultados No Dominados de MOSCA2b.²²

Como conclusión, los resultados obtenidos con el algoritmo MOSCA2b son parecidos a los obtenidos con el algoritmo MOGLS.

¹⁷ Borbulya, István. "An EC-Memory based Method for the Multi-Objective TSP", Proceedings of the 9th annual conference on Genetic and evolutionary computation, New York, 2007.

¹⁸ Borgulya, István. "An Evolutionary Algorithm for the bi-objective QAP", Computational Intelligence, Theory and Applications Advances, springer series, part 22, pag. 577-586, 2006.

Las técnicas que se utilizan en la actualidad para resolver problemas combinatorios multi-objetivo tienen problemas en los tiempos de respuestas y la sensibilidad de las variables de salida respecto a los parámetros es de vital importancia para poder conseguir los óptimos globales, por lo que dichos parámetros deben ser analizados antes de establecerlos dentro del algoritmo.

La metaheurística basadas en autómatas finitos determinísticos para la optimización multi-objetivo de problemas combinatorios planteada por Niño, Ardila, Jabba, Donoso en el año 2010¹⁹, es de gran utilidad para encontrar el espacio de soluciones factibles. La cota de crecimiento del número de estados de Q es exponencial respecto al número de variables de decisión. La solución encontrada por la metaheurística trata de no caer en óptimos locales con lo que evita gran número de iteraciones del programa, aproximando el conjunto de estados Q_ϕ de tal forma que contengan las soluciones óptimas del total de estados Q a partir de Q_0 . Los resultados de la metaheurística MIDA fueron comprobados para el Problema del Agente Viajero (TSP por sus siglas en inglés) y se observa que tuvo mejores resultados que otras técnicas. A continuación se muestran los resultados obtenidos al compararla con las técnicas BCA, PACO, CA, BCM, MOAQ, MACS, MONA, NSGA-II, USBC y SPEA2 con métricas GD, IGD, GVND y ESPACIAMIENTO para diferentes instancias.

¹⁹ Niño, Elías; Ardila, Carlos; Jabba, Daladier; Donoso, Yesid. “A Novel Algorithm Base On Deterministic Finite Automaton for Solving the Mono-Objective Symmetric Traveling Salesman Problem”, International Journal of Artificial Intelligence, vol. 5, No A10. pag. 510, North America, 2010.

| | | MÉTRICA | | | |
|---------|-------------|---------------|---------------|-------------|---------------|
| TÉCNICA | | GD | IGD | GVND | ESPACIAMIENTO |
| | BCA | 566.924 | 107.864.786 | 80 | 0.0462 |
| | PACO | 1.872.472 | 294.171.143 | 40 | 0.1680 |
| | CA | 1.438.240 | 486.924.501 | 67 | 0.0588 |
| | BCM | 772.998 | 200.533.008 | 80 | 0.1907 |
| | MOAQ | 1.201.045 | 414.220.084 | 74 | 0.0840 |
| | MIDA | 0.0041 | 0.5795 | 8013 | 0.0012 |
| | MACS | 271.489 | 64.317.973 | 129 | 0.0452 |
| | MONA | 2.807.074 | 301.220.496 | 27 | 0.2460 |
| | NSGA-II | 2.471.434 | 648.593.292 | 45 | 0.1378 |
| | USBC | 163.255 | 47.317.026 | 184 | 0.0302 |
| | SPEA2 | 2.283.903 | 630.212.864 | 48 | 0.2007 |

Tabla 3- Comparación de Resultados de las Técnicas BCA, PACO, CA, BCM, MOAQ, MACS, MONA, NSGA-II, USBC, SPEA2 y MIDA Utilizando Métricas para la Instancia KROAB100

| | | MÉTRICAS | | | |
|---------|-------------|---------------|------------------|-------------|---------------|
| TÉCNICA | | GD | IGD | GVND | ESPACIAMIENTO |
| | BCA | 132.641 | 21.731.927 | 77 | 0.1150 |
| | PACO | 630.131 | 119.450.227 | 38 | 0.3266 |
| | CA | 442.144 | 97.786.389 | 49 | 0.0969 |
| | BCM | 305.382 | 95.200.810 | 70 | 0.1634 |
| | MOAQ | 275.772 | 47.927.332 | 55 | 0.1362 |
| | MIDA | 0.0322 | 1.151.132 | 7302 | 0.0005 |
| | MACS | 65.284 | 12.154.514 | 117 | 0.1019 |
| | MONA | 543.892 | 124.798.477 | 45 | 0.1398 |
| | NSGA-II | 284.399 | 69.019.911 | 64 | 0.2106 |
| | USBC | 63.924 | 12.879.326 | 123 | 0.0401 |
| | SPEA2 | 587.052 | 114.876.566 | 40 | 0.2946 |

Tabla 4- Comparación de Resultados de las Técnicas por BCA, PACO, CA, BCM, MOAQ, MACS, MONA, NSGA-II, USBC, SPEA2 y MIDA Utilizando Métricas para la Instancia KROAB50

| | | MÉTRICA | | | |
|---------|---------|---------------|---------------|-------------|---------------|
| | | GD | IGD | GVND | ESPACIAMIENTO |
| TÉCNICA | BCA | 524.217 | 132.613.962 | 94 | 0.0358 |
| | PACO | 2.036.702 | 293.610.403 | 36 | 0.2220 |
| | CA | 1.620.395 | 465.911.182 | 57 | 0.0539 |
| | BCM | 650.868 | 151.798.542 | 81 | 0.0510 |
| | MOAQ | 1.116.502 | 352.935.484 | 72 | 0.1238 |
| | MIDA | 0.0000 | 0.0000 | 5806 | 0.0020 |
| | MACS | 343.191 | 89.566.726 | 118 | 0.0841 |
| | MONA | 3.427.103 | 310.464.039 | 22 | 0.3837 |
| | NSGA-II | 2.326.861 | 653.203.249 | 47 | 0.2000 |
| | USBC | 168.676 | 54.337.462 | 187 | 0.0322 |
| | SPEA2 | 2.671.160 | 623.492.827 | 40 | 0.2047 |

Tabla 5- Comparación de Resultados de las Técnicas BCA, PACO, CA, BCM, MOAQ, MACS, MONA, NSGA-II, USBC, SPEA2 y MIDA Utilizando Métricas para la Instancia KROBC100

| | | MÉTRICA | | | |
|---------|---------|---------------|----------------|-------------|---------------|
| | | GD | IGD | GVND | ESPACIAMIENTO |
| TÉCNICA | BCA | 407.928 | 87.403.558 | 83 | 0.0656 |
| | PACO | 1.513.101 | 235.743.924 | 38 | 0.2389 |
| | CA | 1.166.926 | 350.207.357 | 57 | 0.0698 |
| | BCM | 576.416 | 149.177.453 | 77 | 0.1350 |
| | MOAQ | 864.439 | 271.694.300 | 67 | 0.1147 |
| | MIDA | 0.0121 | 385.642 | 7040 | 0.0013 |
| | MACS | 226.654 | 55.346.404 | 121 | 0.0771 |
| | MONA | 2.259.356 | 245.494.337 | 31 | 0.2565 |
| | NSGA-II | 1.694.231 | 456.938.817 | 52 | 0.1828 |
| | USBC | 131.951 | 38.177.938 | 164 | 0.0342 |
| | SPEA2 | 1.847.372 | 456.194.086 | 42 | 0.2333 |

Tabla 6- Comparación de Promedios de los Resultados de las Técnicas por BCA, PACO, CA, BCM, MOAQ, MACS, MONA, NSGA-II, USBC, SPEA2 y MIDA con las Diferentes Métricas.

2. MARCO TEÓRICO

Muchos de los problemas que encontramos en la industria pueden ser solucionados con programación lineal y no lineal. Con el mejoramiento de las tecnologías utilizadas en el sector, aumenta la cantidad de funciones objetivo que se buscan optimizar, por lo que se hace necesario el desarrollo de nuevas técnicas que den solución a este tipo de problemas. Pero encontrar una solución en donde existe una gran cantidad de variables involucradas, las soluciones en el tiempo polinomial resulta ser cada vez más difícil de encontrar. Para manejar este tipo de inconvenientes se formulan como problemas del tipo no polinomial. Tener la posibilidad de de solucionar problemas multi-objetivo, es una oportunidad de mejoramiento significativo para diferentes sectores de la industria productiva y además de aprovechamiento óptimo de los recursos con los que se cuenta para cada proceso. Técnicas como las metaheurísticas es una forma efectiva de dar respuesta a problemas multi-objetivo, dependiendo de los parámetros de entrada. En el marco teórico se busca mostrar las definiciones que permitan explicar el funcionamiento de la teoría de autómatas finitos deterministas multi-objetivo y las metaheurísticas.

2.1. OPTIMIZACIÓN

Cuando en la industria se presenta un problema, lo primero que hacemos como ingenieros es buscar solucionarlo de la manera más sencilla posible. Luego de ello, el paso lógico a seguir es mejorarlo de tal forma que pueda obtener una solución del problema, pero a la vez optimizarlo.

Para poder optimizar un proceso debe conocerse plenamente. Es decir, manejar todas las variables que están involucradas, los factores que puedan perturbar el proceso, etc. La

optimización no solo nos sirve para que el proceso sea más rápido, sino también para mejorar el uso de otras variables de interés, como el personal o el material necesario para tal proceso.

Un problema de optimización está conformado por:

- **VARIABLES DE OPTIMIZACIÓN:** Las variables de optimización son los valores que se modifican para resolver el problema²⁰.
- **FUNCIÓN OBJETIVO:** Un problema de optimización puede tener una o varias funciones objetivo. Generalmente, estas funciones se expresan en términos de las variables de decisión y es la función que se desea optimizar, por lo cual es aquella en la que se evalúan las posibles soluciones. Cuando se tiene una sola función, se considera que es un problema de optimización mono-objetivo y cuando se evalúan más de dos funciones objetivo se habla de un problema de optimización multi-objetivo²¹.
- **RESTRICCIONES:** Las restricciones se expresan en ecuaciones de igualdad o desigualdad. Para que una solución del problema se considera factible, se deben cumplir con todas las restricciones planteadas en él. En caso de que no tenga ninguna restricción, todas las soluciones de la función objetivo serán válidas para resolver el problema²².

²⁰ Cagnina, Leticia. “Optimización Mono y Multiobjetivo a través de una Heurística de Inteligencia Colectiva”, Facultad de Ciencias Físico Matemáticas y Naturales, Universidad Nacional de San Luis, Argentina, 2010.

²¹ Cagnina, Leticia. “Optimización Mono y Multiobjetivo a través de una Heurística de Inteligencia Colectiva”, Facultad de Ciencias Físico Matemáticas y Naturales, Universidad Nacional de San Luis, Argentina, 2010.

²² Cagnina, Leticia. “Optimización Mono y Multiobjetivo a través de una Heurística de Inteligencia Colectiva”, Facultad de Ciencias Físico Matemáticas y Naturales, Universidad Nacional de San Luis, Argentina, 2010.

Existe la posibilidad de que encontremos varias soluciones la optimizar un problemas, es decir, que encontremos varios óptimos que cumplen con todas las restricciones del proceso. Pero la solución que realmente interesa es la solución óptima global, la cual es aquella solución que es la mejor dentro de todas las soluciones factibles. También existe otro grupo de soluciones llamadas soluciones no factibles, las cuales son todas las soluciones que no logran cumplir con todas las restricciones del problema.

2.1.1. Optimización Mono-Objetivo

Los problemas de optimización más sencillos son aquellos en donde solo se tiene una función objetivo a optimizar y se les llama mono-objetivo. Pueden tener no tener restricciones, pero entonces todas las soluciones optimizarían el proceso. En la vida real encontramos tiene al menos una restricción y que además pueden ser muchas: desde la capacidad de la maquinaria hasta el tiempo de proceso de cada elemento del proceso.

Los problemas de optimización mono-objetivo típicamente son modelados como:

(1)

Sujeto a:

(2)

(3)

Donde $f(X)$ es la función objetivo que será optimizada, el vector X es el conjunto de variables independientes y $H(X)$ y $G(X)$ son las restricciones del problema. Generalmente, siempre se encuentran restricciones de no negatividad en los casos productivos, con lo cual se evitan óptimos no posibles en la práctica²³.

Existen tres conjuntos de soluciones que son de interés de estudio:

- **Conjunto universal:** Se llama conjunto universal a las soluciones compuestas por todos los posibles valores que el vector de variables independientes X puede tomar sean o no factibles.
- **Conjunto de soluciones factibles:** Se llama conjunto de soluciones factibles a todos los valores del vector de variables independientes X que cumplen tanto con la restricción $H(X)$, como con la restricción $G(X)$.
- **Conjunto de soluciones óptimas:** Se llama conjunto de soluciones óptimas a todos los valores de X , que además de ser factible optimiza la función objetivo $f(X)$, ya sea en un intervalo específico o de manera global.

En la figura 4 se muestran los diferentes conjuntos de soluciones.

²³ Cagnina, Leticia. “Optimización Mono y Multiobjetivo a través de una Heurística de Inteligencia Colectiva”, Facultad de Ciencias Físico Matemáticas y Naturales, Universidad Nacional de San Luis, Argentina, 2010.

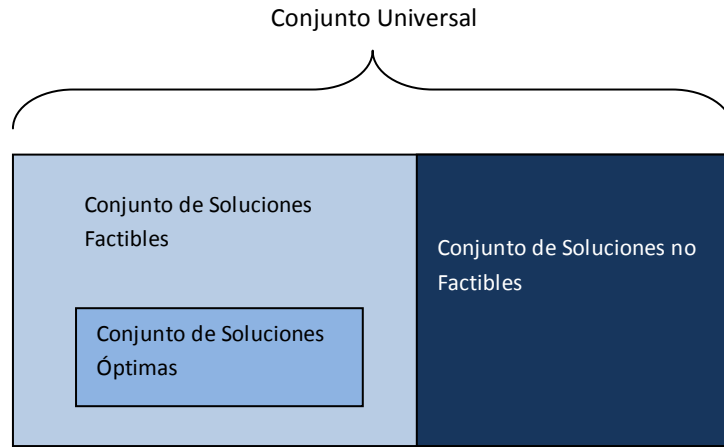


Figura 4- Conjuntos soluciones para un problema mono-objetivo

2.1.2. Optimización Multi-Objetivo

Para poder solucionar problemas mucho más complejos de la industria, se necesita hacer uso de la optimización multi-objetivo. La optimización multi-objetivo se refiere a la solución de problemas en los que se tienen más de una función objetivo con una cantidad determinada de restricciones.

Los problemas de optimización multi-objetivo típicamente son modelados como:

(4)

Sujeto a:

(5)

(6)

Donde $F(X)$ es el conjunto de funciones objetivo a optimizar, $f_1(X)$ la función objetivo 1, $f_2(X)$ la función objetivo 2, $f_n(X)$ la función objetivo n , el vector X es el conjunto de variables independientes y $H(X)$ y $G(X)$ son las restricciones del problema. Generalmente, siempre se encuentran restricciones de no negatividad en los casos productivos, con lo cual se evitan óptimos no posibles en la práctica²⁴. Dado que en el problema a optimizar hay más de una función objetivo, una solución que es óptima para una función objetivo, puede que no sea óptima para otra función objetivo, por lo que la dificultad para encontrar el óptimo global aumenta a medida que aumentan las funciones objetivo.

2.1.2.1. Frente de Pareto

La manera de solucionar la optimización de un problema multi-objetivo es más complicada que la solucionan problemas mono-objetivo. Es por ello que las técnicas son diferentes y cada vez más sofisticadas con el fin de encontrar una solución óptima para cada tipo de problema encontrado. Cuando se busca optimizar un problema mono-objetivo solo hay una función a evaluar y saber si una solución factible es el óptimo global, se resume en evaluar los valores obtenidos factibles dentro de la función objetivo y observar cual solución la optimiza sin olvidar que hay casos en los que es posible que el óptimo se alcance en distintos puntos.

Supongamos que se tiene un problema de optimización mono-objetivo con función objetivo $\min f(x)$. Al evaluar los resultados el conjunto de soluciones obtenido se muestra en la figura 5. Como lo que se pretende es minimizar la función objetivo, la solución óptima será f_1 , dado que en la solución 1 la función toma el menor valor.

²⁴ Cagnina, Leticia. “Optimización Mono y Multiobjetivo a través de una Heurística de Inteligencia Colectiva”, Facultad de Ciencias Físico Matemáticas y Naturales, Universidad Nacional de San Luis, Argentina, 2010.

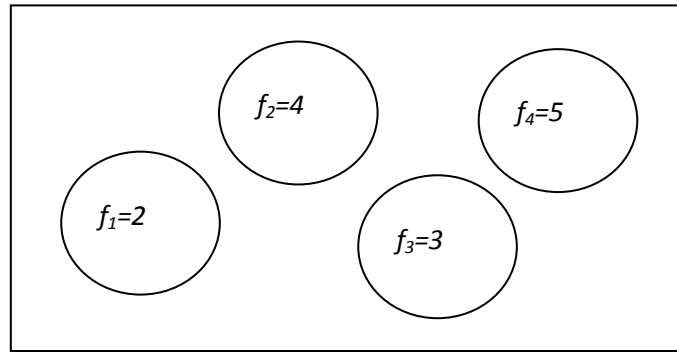


Figura 5- Conjunto de soluciones mono-objetivo.

En el caso de tener un problema de optimización multi-objetivo, la solución no resulta ser tan sencilla, dado que hay varias funciones objetivo y puede que para una de ellas se el óptimo pero para las otras no. Ahora supongamos que se tiene un problema de optimización multi-objetivo con función objetivo $\min F(x)$. Al evaluar los resultados el conjunto de soluciones obtenido se muestra en la figura 6. No es posible determinar si la solución 1 es mejor que la solución 3, ya que $f_{11} > f_{31}$ pero $f_{12} < f_{32}$. Esto no quiere decir que la solución 2 minimiza la función objetivo, solo que las soluciones 1 y 3 son mejores soluciones que la solución 2 dado que son menores, pero no es posible determinar si una es mejor que la otra. Se dice entonces que la solución 2 es dominada por las soluciones 1 y 3 y que además las soluciones 1 y 3 son soluciones No Dominadas

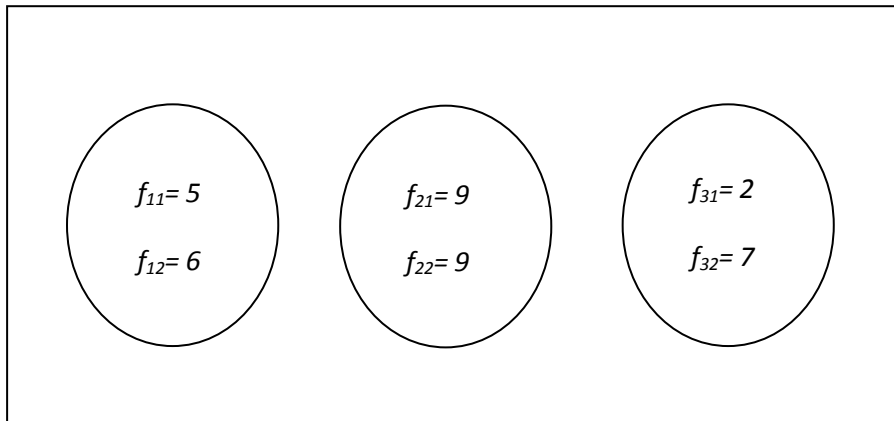


Figura 6. Conjunto de soluciones multi-objetivo.

Se dice que una solución **domina a** una solución si y solo si:

(7)

Formalmente, se dice que una solución **no domina a** una solución si y solo si:

(8)

En el caso de los problemas de optimización multi-objetivo, el conjunto interés de estudio es el conjunto de soluciones no dominadas²⁵, es decir aquellas que optimizan las funciones objetivo. Es posible que para un problema particular más de una solución puedan optimizar las funciones objetivo.

Cuando se tienen problemas con dos o tres objetivos, el conjunto de soluciones no dominadas toman la forma de una gráfica conocida como **Frente de Pareto (FP)**²⁶ como se muestra en la figura 7.

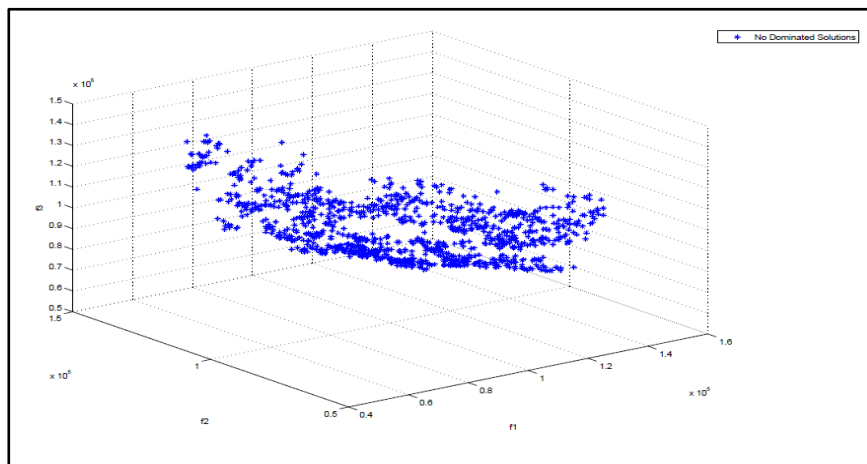


Figura 7. Frente de Pareto tri-objetivo

²⁵ Donoso, Yezid; Fabregat, Ramón. “Multi-Objective Optimization in Computer Networks Using Metaheuristics”, pag. 1 – 3, Aurbach Publications, New York, United States 2007.

²⁶ Figuera, Pau. “Optimización de productos y procesos industriales”, cap. 1, Ediciones Gestión 2000, Barcelona, España, 2006.

2.1.2.1.1. Métricas

Las métricas son utilizadas para medir la calidad de soluciones no dominadas, estas pueden ser singulares ó en contraste con otro conjunto solución. La calidad de las soluciones dependerá del tipo de métrica utilizado. A continuación se mostraran algunas métricas más importante y relevantes.

2.1.2.1.1.1. Generación de Vectores No Dominados

La métrica Generación de Vectores no Dominados (GVND), nos indica cuantos elementos hay en el frete de pareto (FP)²⁷:

(9)

2.1.2.1.1.2. Distancia Generacional

El concepto de métrica Distancia Generacional (DG) fue introducido por Van Veldhuizen como método para estimar qué tan lejos se encuentran los elementos. Es decir que indicara la distancia del FP Aproximado () al FP Real (), esto es²⁸:

²⁷ Tovar, Luis; Coroneal, Margarita; Donoso, Yesid. “Multi Optimización Multiobjetivo en Redes Ópticas con Transmisión Multicast, Utilizando Algoritmos Evolutivos y Lógica Difusa”, Ingeniería & Desarrollo, Número 21, Enero-Junio 2007.

²⁸ Cruz, Nareli; Coello, Carlos. “Un Sistema Inmune Artificial para Solucionar Problemas de Optimización Multiobjetivo”, Grupo de Computación Evolutiva Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2003.

$$\frac{d_i}{\sum_{i=1}^n d_i} \quad (10)$$

Donde d_i es la mínima distancia Euclidiana entre el i -ésimo elemento del FP aproximado con los elementos del FP real.

2.1.2.1.1.3. Distancia Generacional Inversa

La métrica Distancia Generacional Inversa (DGI), nos indica qué tan lejos está el FP de un FP de Referencia, esto es²⁹:

$$\frac{d_i}{\sum_{i=1}^n d_i} \quad (11)$$

Donde d_i es la mínima distancia Euclidiana entre el i -ésimo elemento del FP con los elementos del FP de Referencia.

2.1.2.1.1.4. Espaciamiento

La métrica Espaciamiento (S), nos indica qué tan bien están distribuidas las soluciones sobre el FP. Matemáticamente se define como³⁰:

²⁹ Cruz, Nareli; Coello, Carlos. “Un Sistema Inmune Artificial para Solucionar Problemas de Optimización Multiobjetivo”, Grupo de Computación Evolutiva Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2003.

(12)

Donde \bar{x} con σ y σ^2 es el promedio de todos los x_i .

Para la métrica S un valor 0 se traduce en que todos los elementos de X están equidistantemente espaciados. Por lo tanto, entre menor sea el valor de S , mejor es la distribución de los elementos del conjunto X .³¹

2.1.2.2. Técnicas Clásicas

Existen diversas técnicas clásicas de solución para la optimización multi-objetivo partiendo de soluciones de problemas de optimización mono-objetivo, de las cuales se destacan los siguientes métodos.

2.1.2.2.1. – Restricción

El método de – Restricción consiste en crear un modelo mono-objetivo. El problema multi-objetivo se transforma de la siguiente manera: se toma una de las funciones objetivo

³⁰ Tovar, Luis; Coroneal, Margarita; Donoso, Yesid. “Multi Optimización Multiobjetivo en Redes Ópticas con Transmisión Multicast, Utilizando Algoritmos Evolutivos y Lógica Difusa”, Ingeniería & Desarrollo, Número 21, Enero-Junio 2007.

³¹ Cruz, Nareli; Coello, Carlos. “Un Sistema Inmune Artificial para Solucionar Problemas de Optimización Multiobjetivo”, Grupo de Computación Evolutiva Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2003.

como única función objetivo y el resto de funciones objetivo se plantean como restricciones del problema. Así pues las ecuaciones 5 y 6 quedarán igual, mientras que la ecuación 4 se redefine como³²:

(17)

Sujeto a:

(18)

(19)

(20)

Nótese que la función es la única función objetivo a optimizar en el problema, mientras que las otras funciones objetivo pasaron a ser restricciones del problema. Para cada existe un valor que la acota. Cuando es alterado algún se obtienen nuevas soluciones para la ecuación 17.

2.1.2.2.2. Suma con Pesos

El método de suma de pesos consiste en crear un modelo mono-objetivo, ponderando pesos en las n funciones objetivo del problema. Así pues las ecuaciones 5 y 6 quedarán igual, mientras que la ecuación 4 se redefine como³³:

³² López, Antonio. “Diseño de un Algoritmo Evolutivo Multiobjetivo Paralelo”, Grupo de Computación Evolutiva Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2005.

(13)

Sujeto a:

(14)

(15)

(16)

La ecuación 13 nos permite observar que $F'(X)$ es una combinación lineal de $F(X)$ dada la restricción 14.

2.2. OPTIMIZACIÓN COMBINATORIA

Se le llama optimización combinatoria es una rama de la optimización en matemáticas aplicadas y ciencias de la computación. Es ampliamente utilizada en la investigación de operaciones y otros campos no menos importantes como la inteligencia artificial. Con el estudio de la optimización combinatoria ha sido posible explorar el espacio de soluciones de problemas multi-objetivo usualmente considerados de gran dificultad, logrando reducir el tamaño del conjunto de soluciones y mejorando la eficiencia de programaciones basados

³³ López, Antonio. “Diseño de un Algoritmo Evolutivo Multiobjetivo Paralelo”, Grupo de Computación Evolutiva Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2005.

en esta técnica. Generalmente, se relaciona la optimización combinatoria con problemas del tipo NP (No Polinomial) y NP-hard, en donde mejora la solución de problemas como el del agente viajero por ejemplo³⁴. El Problema del Agente Viajero (Traveling Salesman Problem TSP), considera un viajero que partiendo de una ciudad debe partir a un conjunto de ciudades y volver a la ciudad de origen, pasando una vez por cada ciudad. Lo que se busca es minimizar el costo del recorrido por todas las ciudades y el problema se modela de la siguiente forma:

(21)

Sujeto a:

(22)

(23)

(24)

El TSP es uno de los problemas más estudiado dado que sirve de planteamiento de otros problemas dentro diferentes áreas tales como la computación, la transmisión y distribución de energía eléctrica, manufacturación de productos, minería, entre otros, donde cada vez

³⁴ Quezada, Fátima; Padilla, Felipe; De León, Eunice; Caballos, Rebeca. “Sistema Inmune Artificial para Resolver el Problema del Agente Viajero”, Segundo Congreso Estatal “La Investigación en el Posgrado”, México, Noviembre, 2010.

hay más funciones objetivo a optimizar y más variables y restricciones involucradas dentro de los procesos.

Los problemas combinatorios suelen clasificados en dos grupos dependiendo de la existencia o no de algoritmos polinomiales:

- **Problema tipo polinomial (P):** Es un tipo de problema que no resulta ser muy difícil de resolver y existe gran variedad de herramientas computacionales que ayudan a encontrar el óptimo global del problema.
- **Problema tipo no- polinomial (NP):** En este tipo de problemas encontrar el conjunto de soluciones es más complicado, por lo que no es común encontrar herramientas computacionales que ayude a encontrar la solución óptima.

2.2.1. Técnicas Heurísticas

Las técnicas heurísticas son algoritmos que dan como resultado soluciones aproximadas para problemas combinatorios complejos del tipo NP, donde dichas soluciones son eficientes en la resolución de problemas en procesos. Como ventaja, los algoritmos derivados de heurísticas suelen ser más fáciles de implementar con esfuerzos computacionales significativamente pequeños, aunque no encuentran óptimos globales desde el punto de vista teórico y en problemas de gran tamaño es raro que lo haga. Aún así, las técnicas heurísticas han sido de gran utilidad para resolver problemas combinatorios.

Una heurística busca por medio de una estrategia organizada localizar diferentes soluciones que se vayan acercando al óptimo y están inspiradas en el proceso evolutivo natural³⁵.

2.2.2. Técnicas Metaheurísticas

Las técnicas metaheurísticas permiten aproximar el conjunto de soluciones a las soluciones óptimas de un problema combinatorio. Son usadas cuando no es posible implementar un algoritmo o heurística específica que permita dar solución al problema planteado o cuando con los existentes no se obtienen soluciones satisfactorias. Le termino fue introducido en el artículo de la Metaheurística Búsqueda Tabú en 1986³⁶. De igual manera que la heurísticas, las metaheurísticas están inspiradas en procesos evolutivos naturales.

2.2.2.1. Algoritmos Genéticos

Los Algoritmos Genéticos (AG) son herramientas que imitan a la naturaleza con el fin de resolver problemas complejos partiendo del concepto de la evolución. El primer AG fue desarrollado en los años 60s por John Holland junto con un grupo de colegas y alumnos de la universidad de Michigan. En un principio lo desarrolló para resolver problemas de aprendizaje de máquina. Holland analizó el fenómeno de adaptación en sistemas naturales y

³⁵ Castro, Salvador. “Creación de Portafolios de Inversión Utilizando Algoritmos Evolutivos Multiobjetivo”, Grupo de Computación Evolutiva Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2005.

³⁶ Glover, Fred. “Future Paths for Integer Programming and Links to Artificial Intelligence”, Computers and Operations Research, vol. 13, No. 5, pag. 533-349, 1986.

artificiales, con lo que pudo obtener un modelo tentativo para usarlo en un sistema computacional³⁷.

Los AG permiten representar dos interpretaciones:

- **Al nivel del genotipo:** Se refiere a la carga genética heredada por sus antepasados.
- **Al nivel del fenotipo:** Se refiere a las características visibles del individuo.

Los AG ejecutan una búsqueda simultánea en diferentes regiones del espacio factible, realiza una intensificación sobre algunas de ellas y luego explora otros subespacios a través del intercambio de información entre configuraciones. Los mecanismos que utiliza son:

- **Selección:** Los individuos de la población inicial se generan de forma aleatoria y los operadores genéticos trabajan a escala genotípica sobre la representación elegida (binaria, real, entre otros), obteniéndose una nueva generación de la población. El operador genético que permite elegir las configuraciones de la población actual que participará en la generación de las configuraciones de la nueva población y termina después de decidir el número de descendientes de cada configuración de la población actual³⁸.
- **Recombinación:** También llamado crossover, la recombinación es el mecanismo que permite pasar la información genética de los individuos, es decir, la

³⁷ Castro, Salvador. “Creación de Portafolios de Inversión Utilizando Algoritmos Evolutivos Multiobjetivo”, Grupo de Computación Evolutiva Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2005.

³⁸ Castro, Salvador. “Creación de Portafolios de Inversión Utilizando Algoritmos Evolutivos Multiobjetivo”, Grupo de Computación Evolutiva Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2005.

información de un par de cromosomas originales a sus descendientes, por lo que puede saltar de un espacio de búsqueda a otro con lo que genera diversidad genética³⁹.

- **Mutación:** La mutación es el operador secundario y permite realizar la intensificación en un espacio en particular “caminando” a través de vecinos. Lo que hace es intercambiar el valor de un gen de un cromosoma en una población. De manera aleatoria se elige un cromosoma como posible candidato, luego se genera un número aleatorio y si este es menor que el tasa (se elige en el rango [0.001, 0.05]) de mutación ($\rho < \rho_m$), entonces es realizada la mutación⁴⁰.

El algoritmo que resume los pasos a seguir por un AG se muestran en la figura 8.

| |
|--|
| <p><i>Generar la población inicial</i></p> <p><i>Calcular la función de aptitud para cada individuo de la población</i></p> <p><i>Repetir</i></p> <p><i> Aplicar un operador de selección a los individuos de la población</i></p> <p><i> Aplicar el operador genético de recombinación a los individuos de la población</i></p> <p><i> Aplicar el operador genético de mutación a los individuos de la población</i></p> <p><i>Hasta que se cumpla una condición de terminación.</i></p> |
|--|

Figura 8- Algoritmo Genético⁴¹

³⁹ Castro, Salvador. “Creación de Portafolios de Inversión Utilizando Algoritmos Evolutivos Multiobjetivo”, Grupo de Computación Evolutiva Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2005.

⁴⁰ Castro, Salvador. “Creación de Portafolios de Inversión Utilizando Algoritmos Evolutivos Multiobjetivo”, Grupo de Computación Evolutiva Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2005.

⁴¹ Castro, Salvador. “Creación de Portafolios de Inversión Utilizando Algoritmos Evolutivos Multiobjetivo”, Grupo de Computación Evolutiva Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2005.

2.2.2.2. Recocido Simulado

El Recocido Simulado o Simulated Annealing (SA) es una de las metaheurísticas más clásicas. Su nombre está inspirado en el proceso de recocido del acero y cerámicas que consiste en calentar y luego enfriar lentamente el material para así variar las propiedades físicas del material. Con el calor se aumenta la energía en los átomos para poder desplazar las posiciones iniciales (que sería un mínimo local de energía), y el enfriamiento lento les da mayores probabilidades de recristalizar en configuraciones con menor energía que la energía inicial (que sería el mínimo global). Dada su simplicidad y buenos resultados para diversos tipos de problemas, es una herramienta ampliamente utilizada en diversas áreas. Está basado en el trabajo de 1953 de Metrópolis en el campo de la termodinámica estadística⁴².

El Algoritmo Metrópolis en un principio fue modelado por Metrópolis en 1953 como método de estudio de la termodinámica en estadística. Metrópolis modeló el proceso de recocido simulando los cambios energéticos en un sistema de partículas conforme decrece la temperatura, hasta que converge a un estado estable⁴³. Según las leyes de la termodinámica a una temperatura t la probabilidad de un incremento energético de magnitud dE se puede aproximar a:

(25)

Donde k es la constante física de Boltzmann.

⁴² Díaz, Adenso. “Recocido Simulado”, Universidad de Oviedo, 2004.

⁴³ Díaz, Adenso. “Recocido Simulado”, Universidad de Oviedo, 2004.

Surgió como consecuencia de la comparación de los problemas formulados en el campo de la termodinámica con los campo de la investigación de operaciones. El Algoritmo Metrópolis está basado en el “Método Monte-Carlo”, el cual estudia las propiedades de equilibrio en el análisis del comportamiento microscópico de los cuerpos. Metrópolis genera una secuencia de estados de un sólido, es decir, dado un sólido en un estado i con energía E_i , se genera el siguiente estado j mediante la aplicación de un mecanismo que lo transforma al siguiente estado a través de un pequeño disturbio. Ahora la energía será la energía del siguiente estado, o sea, E_j . Entonces si la diferencia de energía es menor o igual a cero, el estado es aceptado. Si ocurre lo contrario, el estado se acepta con la probabilidad de acuerdo a la ecuación 26.

$$\text{—————} \tag{26}$$

Donde T es la temperatura y k es una constante física conocida como constante de Boltzman.

La ecuación 26 se llama Criterio de Metrópolis. Si la disminución de la temperatura es cambiada de manera paulatina, el sólido alcanzará un estado de equilibrio en cada nivel de temperatura. En el algoritmo es necesario generar un gran número de transiciones en un nivel dado de temperatura, para poder encontrar el equilibrio.

A principios de los años 80s, diversas publicaciones ayudaron en el desarrollo de nuevas herramientas para la solución de problemas combinatorios de gran complejidad, tales como los realizados por Kirkpatrick en 1983⁴⁴ y, tal vez uno de los más importante para el área

⁴⁴ Kirkpatrick, S.; Gellatt, C; Vecchi, M. “Optimization by Simulated Annealing”, Science, 220: pag. 671-680, 1983.

productiva, el hecho por Cerny en 1985⁴⁵ para el TSP, con lo que pudieron mostrar cómo el planteamiento de Metrópolis podría ser aplicado a problemas de optimización, asociando conceptos clave del proceso original de simulación, con elementos de optimización combinatoria. Por lo tanto, cualquier implementación de búsqueda local se puede convertir en una implementación de un recocido simulado al elegir elementos del entorno de manera aleatoria y aceptar los movimientos a una solución no mejor de acuerdo a una probabilidad dada por la ecuación 26. La figura 9 muestra el algoritmo básico del recocido simulado para minimización.

```

Sea f(s) el coste de la solución s y sea N(s) su entorno.
Seleccionar una solución inicial s0;
Seleccionar una temperatura inicial t0 > 0;
Seleccionar una función de reducción de la temperatura α;
Seleccionar un número de iteraciones nrep;
Seleccionar un criterio_de_parada;
REPETIR
REPETIR
  Seleccionar aleatoriamente una solución s ∈ N(s0);
  Sea δ = f(s) - f(s0);
  SI δ < 0 ENTONCES s0 = s
  SINO
  Generar aleatoriamente u ∈ U(0,1);
  SI u < exp(-δ/t) ENTONCES s0 = s;
  FINSINO
HASTAQUE cuenta_iteraciones = nrep
t = α (t);
HASTAQUE criterio_de_parada = CIERTO.

```

Figura 9- Algoritmo Básico del Recocido Simulado para Minimización ⁴⁶

El SA resultado del algoritmo Metrópolis es un algoritmo metaheurístico de búsqueda para problemas de optimización global que trata de encontrar una buena aproximación al óptimo. El algoritmo aplica una acción combinada del mecanismo de generación de alternativas y del criterio de aceptación. El parámetro de control es T_k (la temperatura) y N_k el número de

⁴⁵ Cerny, V. “Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm”, *Journal of Optimization Theory and Applications*, pag. 41—51, 1985.

⁴⁶ Díaz, Adenso. “Recocido Simulado”, Universidad de Oviedo, 2004.

alternativas generadas en la k -ésima iteración del algoritmo. Cuanto T es grande se aceptan grandes deterioros y a medida que T se acerca a cero no acepta ningún deterioro. Esta característica en especial hace que el algoritmo SA sea diferente a otro algoritmo con respecto a los óptimos locales.

A partir del estado i con costo $f(i)$ se genera el estado j con costo $f(j)$. El criterio de aceptación determina si este nuevo estado es aceptado con la siguiente probabilidad:

(27)

La estrategia seguida en SA es partir de una temperatura alta, permitiendo aceptar soluciones de pobre calidad. Luego se disminuye la temperatura y a la vez la posibilidad de aceptar las soluciones peores. Se empieza con una temperatura inicial T_0 y una velocidad de enfriamiento. La temperatura $T_{(k+1)}$ se calcula a partir de T_k y la velocidad de enfriamiento después de haber hecho $N(T_k)$ iteraciones en la temperatura T_k ⁴⁷.

2.2.2.3. Colonia de Hormigas

El primer algoritmo desarrollado en el área de Colonia de Hormigas o Ant Colony Optimization (ACO), fue el Ant System (AS), en donde se estudian sistemas artificiales que simulan colonias de hormigas reales. El ACO se utiliza para resolver problemas de optimización combinatoria, planteados como problemas donde el objetivo es encontrar una

⁴⁷ Díaz, Adenso. “Recocido Simulado”, Universidad de Oviedo, 2004.

secuencia óptima de sus elementos. El ACO ha sido de gran utilidad en la planeación de horarios y ruteo en el TSP. EL algoritmo AS desarrollado por Dorigo⁴⁸ también se utiliza para resolver el TSP. La figura 10 muestra el pseudocódigo del algoritmo original del AS.

```
Inicio
For t = 1 to Max_Iter do // Max_Iter es el número de iteraciones
  For k = 1 to m do //m es número de hormigas (agentes)
    Repetir hasta que la hormiga k complete su recorrido
      Seleccionar la siguiente ciudad que se va a visitar
      Calcular la longitud del recorrido de la hormiga k
      Actualizar los niveles de feromona
  Fin
```

Figura 10- Pseudocódigo del Algoritmo Original del As⁴⁹

El ACO imita el comportamiento normal de las hormigas, en donde las actividades de búsqueda son distribuidas entre agentes con capacidades simples que se representan como hormigas que se comportan como las hormigas reales.

Las hormigas son insectos prácticamente ciegos, de ahí el interés de saber cómo es posible que puedan encontrar rutas más cortas entre el hormiguero y a comida y viceversa y además adaptarse a cambios u obstáculo que se presenten en el ambiente como se muestra en la figura 10. Luego de un estudio exhaustivo fue posible establecer que las hormigas rastrean feromonas dejadas por ellas mismas mientras caminan y es así como logran encontrar los caminos más cortos. Las feromonas son el medio que utilizan para intercambiar información entre los agentes sobre las rutas. Así pues, si una hormiga se

⁴⁸ Dorigo, M.; Maniezzo, V.; Colomi, A. "The Ant System: Optimization by a Colony of Cooperating Agents", IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1):29-41, 1996.

⁴⁹ Mendoza, Benito. "Uso del Sistema de la Colonia de Hormigas para Optimizar Circuitos Lógicos Combinatorios", Universidad Veracruzana, México, 2001.

desplaza del punto A a un punto B deja un rastro de feromonas en el suelo y es como marcan el camino que tomaron. Las hormigas prefieren tomar caminos con alta feromona, con lo que se explica el porqué son capaces de adaptarse a cambios en el entorno. Cuando una hormiga encuentra un obstáculo en el camino de feromonas, ya no lo podrá seguir por lo que deben elegir un nuevo camino, ir a la derecha o a la izquierda del obstáculo como se muestra en cuadro 3 de la figura 11. La elección del camino es de manera aleatoria, pero la probabilidad es que la mitad de las hormigas tomará la derecha y la otra mitad la izquierda como se muestra en el cuadro 4 de la figura 11. En este punto las hormigas eligieron de forma aleatoria el camino más corto para llegar a la comida y después de un tiempo crearán un depósito de feromona más fuerte que el dejado por las hormigas que eligieron el camino más largo, por lo que pasarán más hormigas por el camino corto, ya que saben que llegan al otro lado más rápido que las demás. Como a medida que pasan más hormigas que encontraron el camino corto, más grande será el depósito de feromonas, pasado el tiempo todas las hormigas que vienen atrás preferirán caminar por esta ruta como lo muestra el cuadro 5 de la figura 11⁵⁰.

En el caso que una hormiga vague sola en forma aleatoria, si encuentra un rastro de feromonas, esta puede decidir con alta probabilidad seguirlo y aumentar el nivel de feromonas y hacerlo más fuerte.

En el TSP una hormiga artificial representa al agente que viaja de ciudad en ciudad. El agente escoge la ciudad a la que irá después de acuerdo a una función de probabilidad que dependerá de la cantidad de “feromonas” dejada en ese camino y de una función heurística definida en función de la distancia.

⁵⁰ Mendoza, Benito. “Uso del Sistema de la Colonia de Hormigas para Optimizar Circuitos Lógicos Combinatorios”, Universidad Veracruzana, México, 2001.

Se ubican m hormigas artificiales en ciudades de forma aleatoria, en cada unidad de tiempo ellas se mueven de una ciudad a otra actualizando el rastro de feromonas en los caminos. Esto es llamado una actualización local de rastro. Cuando todas las hormigas completan el recorrido, la hormiga que realizó el camino más corto realiza una nueva actualización de feromonas, pero solo en los caminos que usó, lo que es llamado actualización global de rastro y depende del inverso de la distancia recorrida por la hormiga⁵¹.

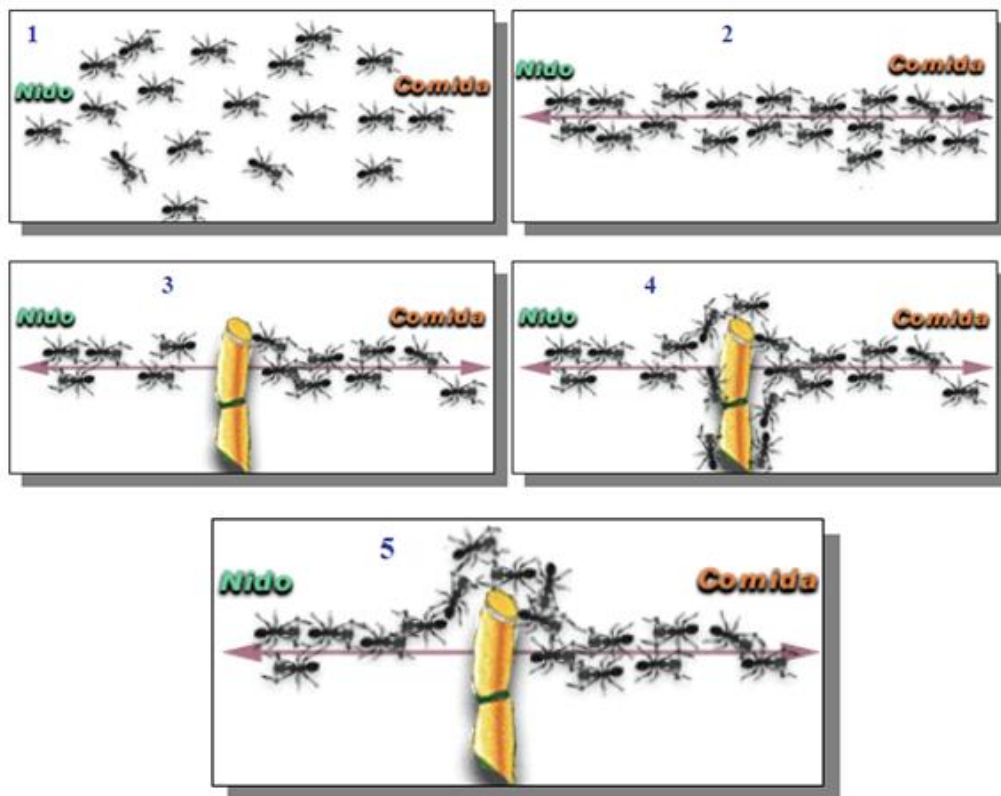


Figura 11- Comportamiento Adaptativo de las Hormigas⁵²

Una hormiga k en la ciudad r , escoge irse para la ciudad s , entre las ciudades que no ha visitado de acuerdo a función de probabilidad:

⁵¹ Mendoza, Benito. “Uso del Sistema de la Colonia de Hormigas para Optimizar Circuitos Lógicos Combinatorios”, Universidad Veracruzana, México, 2001

⁵² Mendoza, Benito. “Uso del Sistema de la Colonia de Hormigas para Optimizar Circuitos Lógicos Combinatorios”, Universidad Veracruzana, México, 2001.

(28)

Donde $\tau(r,u)$ es la cantidad de rastro de feromona en el segmento de viaje (r,u) , $\eta(r,u)$ es una función heurística que se escoge para este caso como el inverso de la distancia entre r y u , α y β son parámetros que pesan la importancia del rastro de feromona y de la función heurística y q es un valor escogido de manera aleatoria entre 0 y 1. s es escogida por medio de la fórmula de probabilidad que favorece aquellas ciudades donde la distancia es más corta y que además tiene un alto nivel de feromona:

(29)

Donde $P_k(r,s)$ es la probabilidad de que la hormiga k e mueva de la ciudad r a la ciudad s .

El rastro de feromonas se actualiza de manera local y global y se hace para recompensar los caminos tomados por el mejor recorrido. Una vez terminado el recorrido por todas las hormigas artificiales, la mejor hormiga deposita una cantidad de feromonas en los caminos visitados que es inversamente proporcional a la longitud del recorrido. Así pues, mientras más corto es el recorrido mayor será la cantidad de feromonas depositados por la hormiga:

(30)

Donde r es un factor entre 0 y 1 que representa la tasa de evaporación de la feromona para evitar que la feromona se acumule infinitamente.

La actualización global es similar a un proceso de aprendizaje reforzado, donde las mejores soluciones tienen prioridad. La actualización local pretende evitar que un camino muy fuerte sea escogido por todas las hormigas, cada vez que una hormiga escoge un camino la cantidad de feromona en el camino es actualizada con la ecuación 30.

(31)

Donde α es un parámetro dado.

2.2.2.4. Búsqueda Tabú

La Búsqueda Tabú (Tabu Search – TS) es un procedimiento metaheurístico que usa memoria adaptativa y estrategias especiales de resolución de problemas. El término Búsqueda Tabú fue introducido por Fred Glover en 1986⁵³ en el mismo artículo donde introdujo el término metaheurística. En 1997 se recopiló todos los resultados de los estudios hechos entre los años 80s y 90s en un libro llamado “*Tabu Search*”, dada su efectividad para resolver problemas de optimización del tipo hard que surgen más que todo en situaciones del mundo real⁵⁴.

La TS es una metaheurística que guía un procedimiento de búsqueda local en la búsqueda de optimalidad global. Lo que hace es realizar una búsqueda a través del espacio de

⁵³ Glover, Fred. “Future Paths for Integer Programming and Links to Artificial Intelligence”, *Computers and Operations Research*, vol. 13, No. 5, pag. 533-349, 1986.

⁵⁴ Melián, Belén; Glover, Fred. “Introducción a la Búsqueda Tabú”. Dpto. de Estadística, I.O. y Computación, Escuela Técnica Superior de Ingeniería Informática, Universidad de La Laguna.

configuraciones delimitando adecuadamente los óptimos locales. La memoria adaptativa del método le permite a la TS implementar procedimientos capaces de realizar la búsqueda en espacios de soluciones eficaces y eficientes. La exploración responsiva del método deriva la suposición de que una mala elección estratégica puede proporcionar más información que una buena elección aleatoria, dado que la mala puede proporcionar elementos útiles sobre cómo guiar la búsqueda hacia zonas prometedoras.

El TS tiene una eficiencia alta ya que evita regresar a los óptimos locales y entrar en un ciclo repetitivo, clasificando los movimientos más recientes como “movimientos tabú”, los cuales prohíben regresar a una configuración ya explorada⁵⁵.

El método tiene dos tipos de memoria:

- **Memoria de corto plazo:** La memoria de corto plazo, como su nombre lo dice, contiene la información de eventos ocurridos recientemente.
- **Memoria de largo plazo:** La memoria de largo plazo almacena los datos de frecuencia de ciertos eventos. Es fundamental para definir las estrategias de diversificación, es decir que permite explorar regiones no visitadas antes.

La TS puede ser caracterizada mediante referencia a la búsqueda por entornos, que para la TS tiene un significado más amplio que en otros métodos. En la figura 12 se muestra el algoritmo de búsqueda en el entorno utilizado por la TS.

⁵⁵ Melián, Belén; Glover, Fred. “Introducción a la Búsqueda Tabú”, Dpto. de Estadística, I.O. y Computación, Escuela Técnica Superior de Ingeniería Informática, Universidad de La Laguna.

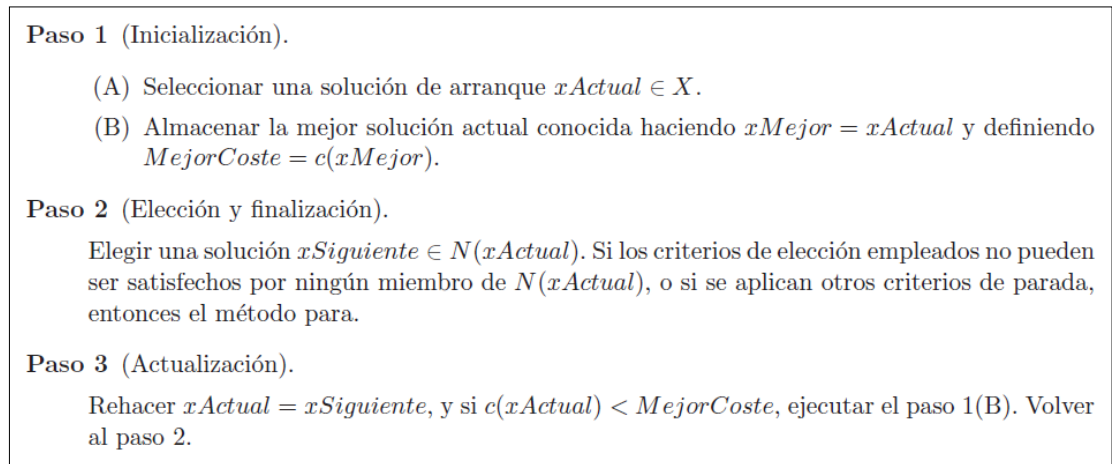


Figura 12- Algoritmo de Búsqueda en el Entorno⁵⁶

Una vez un movimiento ha sido clasificado como “movimiento tabú” y ha sido analizado, se produce una función objetivo mejor que un valor de referencia escogido y se aplica la regla de aspiración que consiste en cancelar la prohibición y aceptar el movimiento. En la figura 13 se puede ver el algoritmo usado por el método TS.

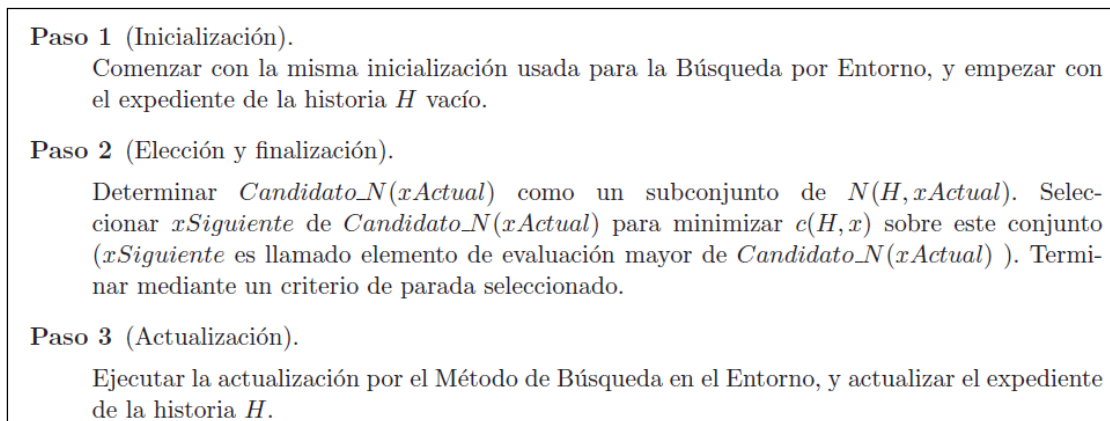


Figura 13- Algoritmo de Búsqueda Tabú

⁵⁶ Melián, Belén; Glover, Fred. “Introducción a la Búsqueda Tabú”, Dpto. de Estadística, I.O. y Computación, Escuela Técnica Superior de Ingeniería Informática, Universidad de La Laguna.

2.2.2.5 Greedy Randomized Adaptive Search Procedure

Los Greedy Randomized Adaptive Search Procedure (GRASP), son algoritmo evolutivo heurístico que tiene como objetivo resolver problemas de la optimización combinatoria del tipo hard. Construye soluciones de alta calidad que son procesadas para obtener aún mejores soluciones. Son algoritmos de tipo iterativo en donde cada iteración incluye una fase de construcción de una solución y otra de pos-procesamiento en la que se optimiza la solución generada en la primera fase⁵⁷. Es útil especialmente en aquellos problemas que usan indicadores de sensibilidad, en donde con dichos factores calcula la variación de la función objetivo con respecto a las variables de interés del problema de optimización y las usa para identificar los atributos atractivos. Su planteamiento está entre el planteado en el Recocido Simulado y la Búsqueda Tabú para el análisis de la fase exploratoria.

La figura 14 muestra el diagrama de flujo del algoritmo GRASP.

⁵⁷ Tupia, Manuel. "Un Algoritmo GRASP para Resolver el Problema de la Programación de Tareas Dependientes en Máquinas Diferentes (Task Scheduling)", Universidad Nacional Mayor de San Marcos, Lima, Perú, 2005.

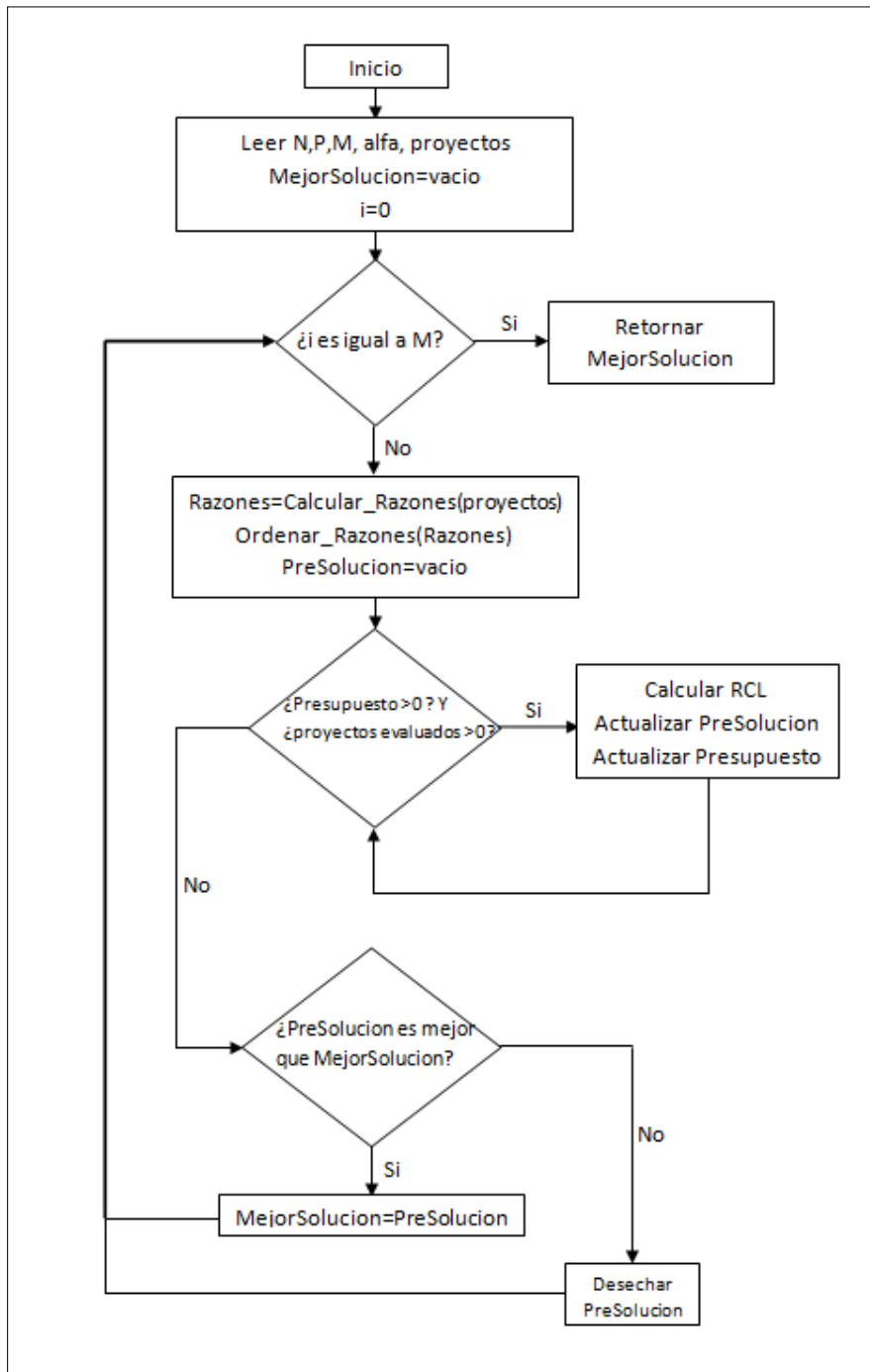


Figura 14- Diagrama de Flujo del Algoritmo GRASP

Las fases a seguir en un algoritmo GRASP son:

- **Implementar una fase de pre-procesamiento:** En esta fase se trata de identificar los atributos más interesantes del problema y con dichos atributos se realiza el proceso de búsqueda. Con esto se logra disminuir el espacio de soluciones que se quiere explorar.

- **Realizar la fase de búsqueda constructiva:** En esta fase se encuentra una solución de calidad para el problema con base en un algoritmo heurístico constructivo donde, en cada paso, se escoge un elemento de una lista de tamaño k denominada *RCL*. Es ahí donde se clasifican las variables más atractivas y se obtiene una incumbente de buena calidad para el problema. En fase de búsqueda constructiva se siguen los siguientes pasos:
 - Escoger una solución inicial que puede ser vacía, es decir, sin adicionar variables la cual se transforma en la solución actual del problema.

 - Para la solución actual del problema, elaborar una lista que clasifica las mejores k variables que identifican el indicador de sensibilidad.

 - Escoger en forma aleatoria o probabilística una de las variables de la lista y actualizar la solución con la adición o sustracción de la variable escogida.

 - Si la solución actual es factible o se satisface el criterio de parada, se finaliza la fase constructiva.

- **Realizar la fase exploratoria y actualizar la mejor solución encontrada si se supera la incumbente:** En esta fase se procura encontrar una solución óptima local en la vecindad de la solución de la fase constructiva.
- **Si el criterio de parada no se satisface volver al paso 2. Si no, finalizar el proceso. La respuesta del algoritmo es la mejor solución almacenada.**

En la figura 15 se puede observar la estructura básica de un algoritmo GRASP.

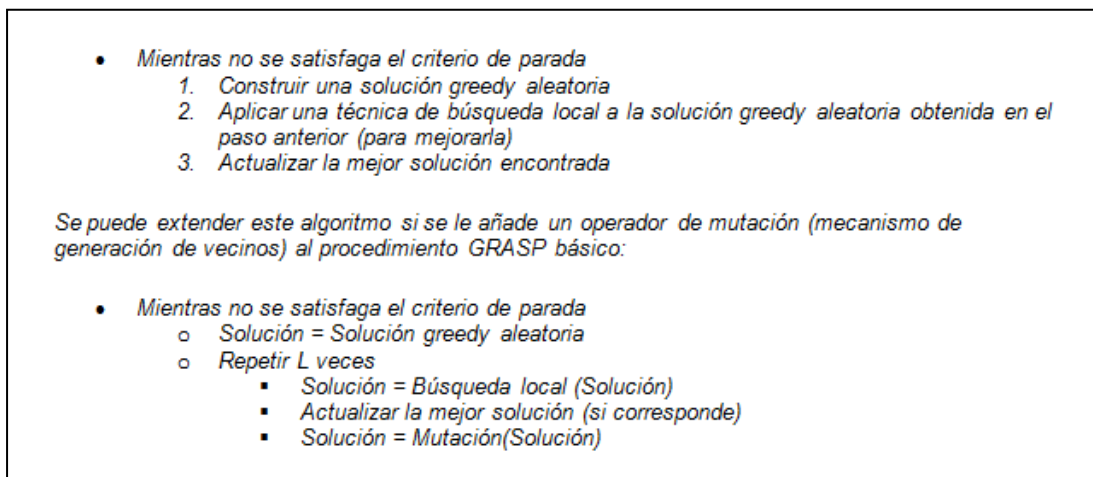


Figura 15- Estructura Básica de un Algoritmo GRASP⁵⁸

⁵⁸ Tupia, Manuel. "Un Algoritmo GRASP para Resolver el Problema de la Programación de Tareas Dependientes en Máquinas Diferentes (Task Scheduling)", Universidad Nacional Mayor de San Marcos, Lima, Perú, 2005.

2.3. AUTÓMATAS

2.3.1. Autómata Finito Determinista⁵⁹

Un Autómata Finito Determinista (AFD), es un dispositivo teórico capaz de procesar una secuencia finita de símbolos de un alfabeto, cambiando su estado si procede. De entre los posibles estados que alcanzar el autómata se distinguen los aceptadores que indican el reconocimiento de la secuencia tratada.

Un AFD está definido como:

(32)

Donde Q es un conjunto no vacío finito de estados, Σ es un alfabeto finito es la función de transición, q_0 es el estado inicial y F , es el conjunto de estados de finalización o aceptadores.

Esta definición es estática y no determina el funcionamiento del autómata. Indica solo los elementos que la conforman y no su comportamiento. Para darle dinamismo es necesario mostrar la definición de la función de transición extendida, con lo cual se podrá decir cuál es el comportamiento que asumirá el autómata de acuerdo al alfabeto de entrada, con la función δ^* dentro del dominio Σ^* .

⁵⁹ Castro, David. "Teoría De Autómatas y Lenguajes Formales y Gramática", Universidad de Alcalá, 2003.

2.3.1.1. Función de Transición Extendida

Dado un autómata finito determinista D , se define la extensión al dominio Σ^* , (denotada por δ^*) de manera inductiva en la longitud de cadenas:

Dado que δ , definimos

Dados q , s y a , definimos

Con esta definición se logra formalizar el funcionamiento de un autómata: dada una cadena w , el AFD lee cada símbolo de entrada, de izquierda a derecha, y va cambiando su estado, si procede en función del estado y el símbolo leído. Las cadenas en que el autómata se encuentra en un estado aceptador conforman el lenguaje del autómata. La función de transición extendida está definida para todos los elementos de Σ^* , con lo que se obliga a todos los elementos que pertenece al conjunto de estados finitos Q a estar interconectados por medio de alguna letra del alfabeto de entrada con otro estado de este mismo conjunto. Es decir, para cualquier letra del alfabeto de entrada:

(33)

2.3.1.2. Lenguaje Aceptado por un AFD

Dado un autómata finito determinista D , se define el lenguaje aceptado por el autómata, denotado $L(D)$, como:

Normalmente cuando se describe un autómata se detalla cada uno de sus elementos.

2.3.1.3. Formas de Representación de los AFDs

2.3.1.3.1. Tablas de Transiciones

La manera en la que usualmente se detalla cada elementos de un AFD, más específicamente la función de transición suele representarse mediante un tabla llamada "tabla de transiciones", la cual indica para cada estado par estado-símbolo, el estado indicado por , como se muestra en la tabla 7 cómo ejemplo. Las filas corresponden a los estados y las columnas a las entradas. El estado inicial se representa señalándolo con una flecha y los finales con un asterisco. El valor correspondiente a la fila cuyo estado es q y a la columna de entrada es el estado $\Lambda(q,)$.

| | 0 | 1 |
|-------------------|-------|-------|
| $\rightarrow q_0$ | q_2 | q_0 |
| $* q_1$ | q_1 | q_1 |
| q_2 | q_2 | q_1 |

Tabla 7- Tabla de Transiciones⁶⁰

2.3.1.3.2. Diagramas de Transición

Otra de las representaciones de AFDs se llama diagramas de transición. En ella se representa el autómata mediante un grafo orientado sujeto a las siguientes consideraciones:

⁶⁰ Castro, David. "Teoría De Autómatas y Lenguajes Formales y Gramática", Universidad de Alcalá, 2003

- Hay un nodo para cada estado de Q .
- Para cada estado q de Q y cada símbolo de entrada a , sea $\delta(q, a) = p$. Entonces el diagrama de transiciones tendrá un arco desde el nodo q al nodo p con la etiqueta a . Si existen varios símbolos de entrada que provoquen una transición del estado q al p , entonces el diagrama de transiciones puede tener solo un arco etiquetado con la lista de esos símbolos.
- Al nodo correspondiente al estado inicial q_0 , llega una flecha sin origen concreto.
- Los nodos correspondientes a los estados de aceptación (finalizadores), están marcados por doble círculo con el fin de distinguirlos de los restantes.

Por ejemplo considérese el autómata $M = (Q, \Sigma, \delta, q_0, F)$, donde $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1\}$, q_0 es el estado inicial y $F = \{q_2\}$. La función de transición se muestra en la tabla 8.

| | 0 | 1 |
|-------------------|-------|-------|
| $\rightarrow q_0$ | q_0 | q_1 |
| q_1 | q_2 | q_1 |
| $*q_2$ | q_2 | q_2 |

Tabla 8- Tabla de Transiciones del Ejemplo

El diagrama de transiciones resultante para este ejemplo se muestra en la figura 16.

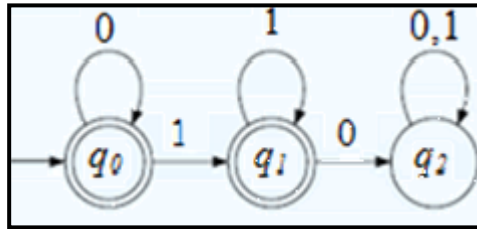


Figura 16- Diagrama de Transiciones del Ejemplo

2.3.2. Autómata Finito Determinista de Intercambio⁶¹

Un Autómata Finito Determinista de Intercambio (AFDI), es una gráfica con la que es posible representar el espacio de soluciones factibles de un problema de naturaleza combinatoria, sin repeticiones y donde el orden de los elementos importa. Se define como:

(34)

Donde π es el vector que contiene el orden original de los elementos asociados al problema representado, f es la función que se desea optimizar en el problema modelado, Q es el conjunto de estados que conforman el autómata, cada estado representa una solución al problema modelado. La estructura de cada estado viene conformada de la siguiente manera:

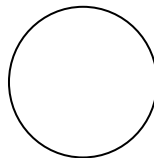


Figura 17- Estructura de un estado de un AFD – I

⁶¹ Niño, Elias; Ardila, Carlos, “Algoritmo Basado en Autómatas Finitos Deterministas para la Obtención de Óptimos Globales en Problemas de Naturaleza Combinatoria”, Revista de Ingeniería y Desarrollo, No 25. pp 100 – 114, ISSN 0122 - 3461.

Donde σ corresponde a un orden específico de los elementos asociados al problema modelado y así mismo se convierte en una solución factible al problema propuesto.

Σ es el alfabeto finito de entrada. Este contiene todas las posibles combinaciones entre los índices del vector de entrada agrupándolos en parejas, debido a que estos son distintos, las parejas de Σ vendrían dadas por tuplas que cumplen la condición:

$$(35)$$

Donde n es el número de elementos del vector de entrada.

Definimos δ como la función de transición la cual toma un estado q , σ , y una tupla de Σ y devuelve un nuevo estado q' , esto es $\delta(q, \sigma) = q'$, en donde $q, q' \in Q$. Es importante resaltar que, debido a que los AFD – I provienen de los AFD (de acuerdo a los diagramas de transición), cada estado $q \in Q$ debe tener transición con todos los símbolos de Σ , q_0 es el estado inicial del AFD – I. Este contiene una solución inicial para el problema modelado, F es el conjunto de estados de finalización, para los AFD – I todos los elementos de F son elementos de Q .

Ejemplo 2: supóngase el vector de entrada $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$ y la función objetivo:⁶²

$$(36)$$

⁶² Niño, Elías; Ardila, Carlos, “Algoritmo Basado en Autómatas Finitos Deterministas para la Obtención de Óptimos Globales en Problemas de Naturaleza Combinatoria”, Revista de Ingeniería y Desarrollo, No 25. pp 100 – 114., ISSN 0122 - 3461.

Para la construcción del AFDI equivalente a estos datos, procedemos a hacer lo siguiente:

- **Establecer el estado inicial:** En este caso nuestro estado inicial contendrá el vector de entrada , por lo tanto .
- **Establecer el conjunto :** Por 34 sabemos que .
- **Establecer la función :** Debido a que los AFDI exigen que los estados tengan transiciones con todos los símbolos del alfabeto, comenzamos por el estado inicial el cual es , cada estado nuevo lo nombramos como , luego hacemos transiciones con cada hasta agotarlos todos como podemos apreciarlo en la tabla 9.

| | | |
|-----------|--|--|
| (2,1,3) = | | |
| (2,3,1) = | | |
| (3,2,1) = | | |
| (1,2,3) = | | |
| (1,3,2) = | | |
| (3,1,2) = | | |

Tabla 9- Función para el AFD I con vector de entrada

Como puede observarse en la tabla 9 todos los estados tienen transiciones con todos los símbolos del alfabeto y en consecuencia el AFD – I contiene todas las posibles permutaciones para los elementos del vector de entrada .

- **Establecer los valores obtenidos para cada evaluación de los en la función objetivo.** El siguiente paso es calcular el valor de la evaluación en la función objetivo de cada , con lo cual obtenemos la tabla 10.

| Estado | Vector | Valor |
|--------|---------|-------|
| | (1,2,3) | 1 |
| | (1,3,2) | 1.1 |
| | (3,2,1) | 1.4 |
| | (2,1,3) | 1.1 |
| | (2,3,1) | 1.3 |
| | (3,2,1) | 1.4 |

Tabla 10- Evaluación de los vectores en la función objetivo

El AFDI de la figura 18 representa el autómata de este ejemplo. Nótese que cada estado tiene transición con todos los elementos de Σ lo cual representa el intercambio llevado a cabo en el vector asociado al estado para alcanzar el nuevo estado cuyo vector es

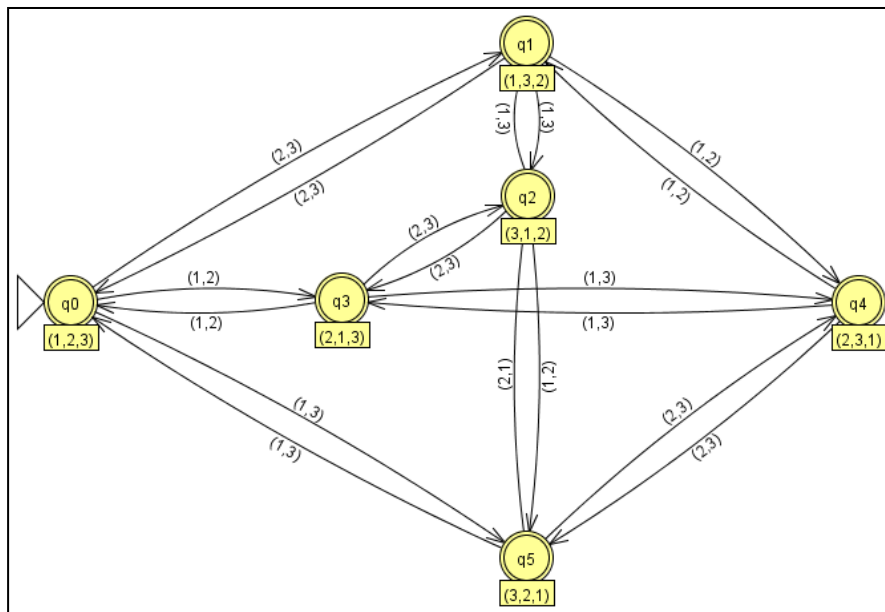


Figura 18- Representación del AFDI para el vector de entrada

2.3.3. Autómata Finito Determinista Multi-Objetivo⁶³

Un Autómata Finito Determinista Multi-Objetivo (AFDM) es una estructura de datos que permite el modelado del espacio de soluciones factibles de un problema combinatorio bajo un esquema multi-objetivo.

Formalmente, un AFDM se define como:

(37)

2.3.3.1. Conjunto de Estados⁶⁴

es el conjunto de Estados del AFDM. Cada estado contiene un vector solución factible del problema combinatorio modelado. contiene un orden de las variables decisión del problema combinatorio.

La estructura de un estado se aprecia en la figura 19.

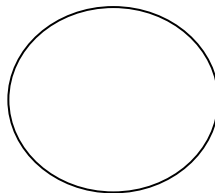


Figura 19- Estructura de un estado .

⁶³ Niño, Elías, "Algoritmo Basado en Autómatas Finitos Deterministas Y Expresiones Regulares para solucionar el Problema del Agente Viajero Simétrico Multi-Objetivo ", VIII Congreso de Investigación Operativa - Optima 2009, Chile, 2009.

⁶⁴ Niño, Elías; Ardila, Carlos; Jabba, Daladier; Donoso, Yesid, "Mods: A Novel Metaheuristic Of Deterministic Swapping For The Multi – Objective Optimization Of Combinatorial Problems ".

No existen dos estados con el mismo . Se define como el vector solución del estado

Ejemplo 1: Sea una empresa que tiene tres máquinas que realizan un trabajo cualquiera, se tienen tres procesos cuya duración es de 10, 50 y 5 minutos respectivamente, si los procesos se ejecutan en paralelo y pueden utilizar cualquiera de las máquinas disponibles, ¿De cuantas maneras se pueden asignar las máquinas a procesos distintos para que se ejecuten?

Fácilmente esta situación puede presentarse a diario en cualquier empresa de producción que trabaje con máquinas y procesos en paralelo, supongamos que el vector representa la solución máquina 1 ejecuta , máquina 2 ejecuta , y máquina 3 ejecuta , por lo tanto el conjunto de estados que representa el conjunto de soluciones factibles asociados a este problema son los representados en la figura 20.

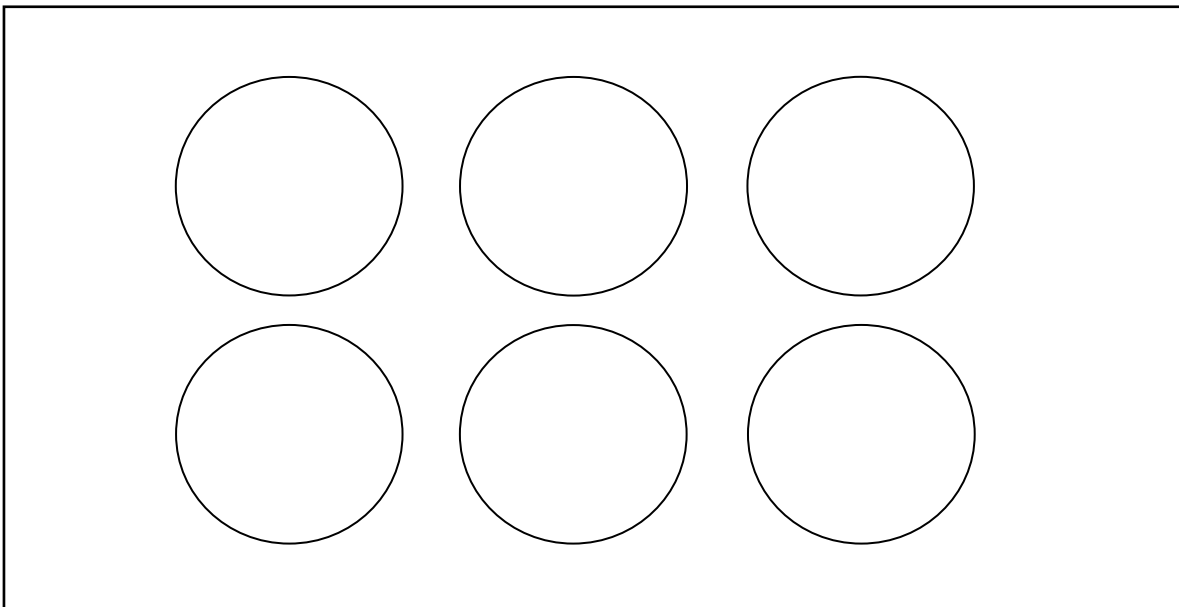


Figura 20- Representación del Espacio de Soluciones Factibles para el Problema del Ejemplo

Como puede apreciarse en la figura 20, por la definición 1 tenemos que :

$$y = \dots$$

Debido a que en cada estado es diferente, y como en un AFDM el número de estados es igual al número de todas las posibles soluciones, si no existen restricciones en el problema, existen estados, esto es:

(38)

2.3.3.2. Alfabeto Finito de Entrada⁶⁵

es el conjunto que se conoce como el alfabeto finito de entrada y está compuesto por los elementos que cumplen la siguiente condición:

(39)

Esto implica que contiene todas las posibles maneras de intercambiar los elementos de un vector. Consecuentemente, indica todas las posibles maneras de intercambiar los elementos de cualquier .

Dado un AFDM cuyos vectores solución son de norma , el número de elementos del conjunto es igual a:

⁶⁵ Niño, Elías; Ardila, Carlos; Jabba, Daladier; Donoso, Yesid, "Mods: A Novel Metaheuristic Of Deterministic Swapping For The Multi – Objective Optimization Of Combinatorial Problems ".

$$\text{_____} \quad (40)$$

Esto se demuestra de la siguiente manera:

Por la definición de subgrupos, si agrupamos los elementos en grupos de a dos ($k = 2$), tendríamos la siguiente expresión:

$$\text{_____} \quad (41)$$

Aplicando la definición de factorial de manera recurrente en la ecuación 40 se obtiene:

$$\text{_____} \quad (42)$$

Cancelando los _____ y reemplazando el valor de $2! = 2$ en la ecuación 41 se obtiene:

$$\text{_____} \quad (43)$$

Y con ello queda demostrado el teorema expuesto en la ecuación 40

Ejemplo 2: se tiene el conjunto _____ para el problema del ejemplo 1 es:

$$(44)$$

La cantidad de elementos concuerda con la ecuación la propuesta por el teorema de la ecuación 40:

(45)

2.3.3.3. Función de transición⁶⁶

La función de transición describe el comportamiento del AFDM. la función permite el desplazamiento entre los diversos estados del AFDM utilizando elementos de , esto es:

(46)

Como puede apreciarse, la función descrita en la ecuación 46, obtiene un elemento de para perturbar los elementos del vector , es decir, intercambiar los elementos del vector y consecuentemente, caer en otro estado con solución .

La función es definida formalmente como:

(47)

Ejemplo 3: Con los datos del ejemplo 1, la función definida para el estado es:

Sobre cada nuevo estado destino obtenido al aplicar la función (en este caso , y), debe aplicarse la misma función con el objetivo de alcanzar nuevos estados. El proceso termina cuando no se obtienen nuevos estados. Una vez se han obtenido todos los estados, se puede afirmar que se ha recorrido el espacio de soluciones factibles del problema modelado por medio del AFDM, lo que equivaldría a una búsqueda exhaustiva.

⁶⁶ Niño, Elías; Ardila, Carlos; Jabba, Daladier; Donoso, Yesid, "Mods: A Novel Metaheuristic Of Deterministic Swapping For The Multi – Objective Optimization Of Combinatorial Problems ".

| Estado origen | Elemento de | | Estado destino |
|---------------|-------------|--|----------------|
| | | | |
| | | | |
| | | | |

Tabla 11- Función para del AFDM del ejemplo 1.

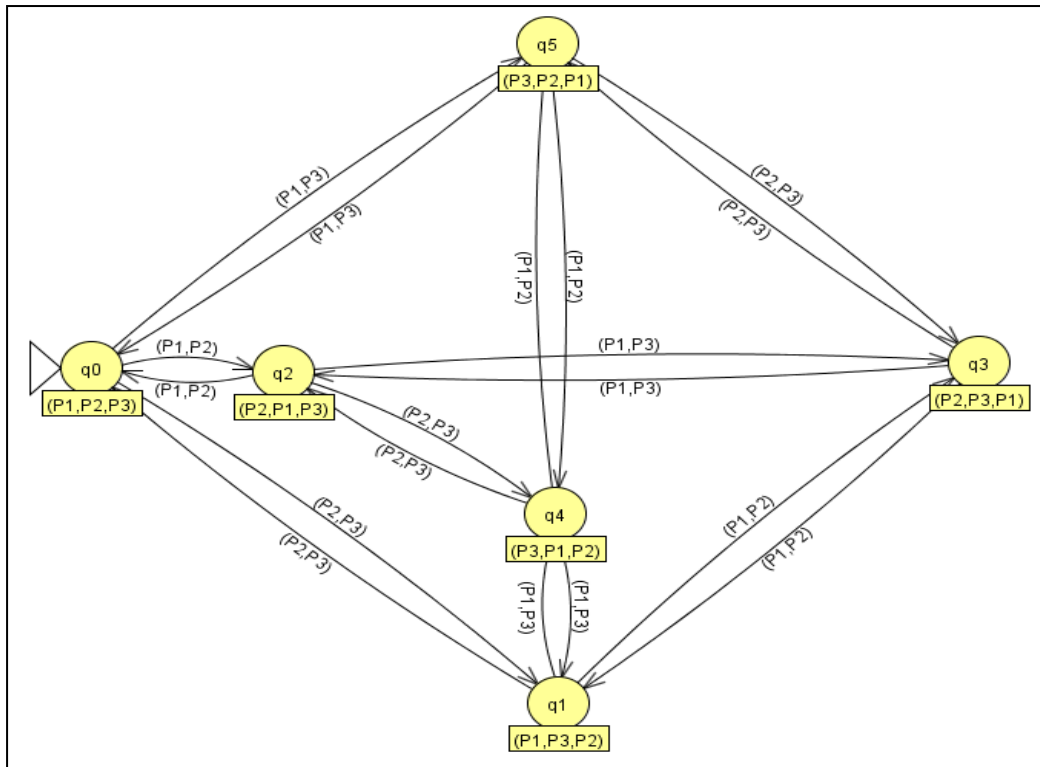


Figura 21- AFDM construido a partir del estado inicial del ejemplo 1.

El autómata de la figura 21 ilustra el autómata construido a partir de la función . La flecha sobre indica que el autómata comenzó a generarse desde ese estado.

2.3.3.4. Conjunto de Estados Iniciales⁶⁷

El conjunto de estados iniciales contiene los estados por el cual se comenzará a generar el autómata. En otras palabras, representa las soluciones iniciales del problema. Así pues en el ejemplo 3 solo contenía un elemento igual al estado .

2.3.3.5. Conjunto de Funciones Objetivos⁶⁸

La función es el conjunto de funciones objetivos del problema, si el problema tiene n objetivos, es:

(48)

Esta función se encuentra definida formalmente como:

(49)

recibe como argumento de entrada una solución . En la ecuación 49, representa el número de variables decisión del problema combinatorio y representa el número de funciones objetivo. Por lo tanto toma las variable decisión y las evalúa en cada función objetivo devolviendo una n – tupla.

⁶⁷ Niño, Elías; Ardila, Carlos; Jabba, Daladier; Donoso, Yesid, "Mods: A Novel Metaheuristic Of Deterministic Swapping For The Multi – Objective Optimization Of Combinatorial Problems ".

⁶⁸ Niño, Elías; Ardila, Carlos; Jabba, Daladier; Donoso, Yesid, "Mods: A Novel Metaheuristic Of Deterministic Swapping For The Multi – Objective Optimization Of Combinatorial Problems ".

Ejemplo 4. Supóngase el siguiente conjunto de funciones bi – objetivo (2 objetivos):

$$- \quad (50)$$

En la ecuación 50 podemos ver que este es un problema bi – objetivo. Si se evalúa el espacio de soluciones factibles del problema 1 en la ecuación 49, se obtienen los resultados de la tabla 12. El FP para este problema puede apreciarse en la figura 22. En este caso en particular recibe el orden en que se asignarán los trabajos en las tres máquinas y devuelve dos valores correspondientes a cada función objetivo.

| Estado | Asignación | | | Tiempos | | | | |
|--------|------------|----|----|---------|----|----|-----|-------|
| | M1 | M2 | M3 | M1 | M2 | M3 | | |
| | P1 | P2 | P3 | 10 | 50 | 5 | 125 | 36.66 |
| | P1 | P3 | P2 | 10 | 5 | 50 | 170 | 29.16 |
| | P2 | P1 | P3 | 50 | 10 | 5 | 85 | 56.66 |
| | P2 | P3 | P1 | 50 | 5 | 10 | 90 | 55.83 |
| | P3 | P1 | P2 | 5 | 10 | 50 | 175 | 26.66 |
| | P3 | P2 | P1 | 5 | 50 | 10 | 135 | 33.33 |

Tabla 12- Función Aplicada a los Estados , , del AFDM del Ejemplo 1

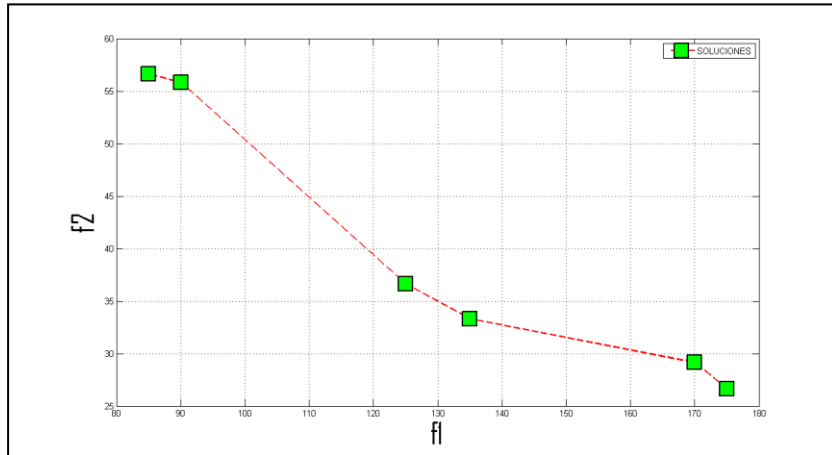


Figura 22- Conjunto de Soluciones Factibles para el Problema del Ejemplo 1

3. METAHEURISTICA PROPUESTA

3.1 METAHEURISTICA BASADA EN AUTOMATAS FINITOS Y ALGORITMOS GENETICOS PARA LA OPTIMIZACION TRI-OBJETIVO DEL PROBLEMA DEL AGENTE VIAJERO SIMETRICO MULTI OBJETIVO

Al mejorar los procesos se crean nuevos retos para optimizar el uso de recursos. Los autómatas finitos deterministas multi-objetivo (AFDM) permiten describir de manera efectiva el espacio de soluciones factibles para problemas combinatorios. Sin embargo, no es posible asegurar que se van a encontrar soluciones óptimas. Esto porque la cota de crecimiento de los estados de Q es del tipo exponencial, proporcional al número de variables de decisión. El espacio de soluciones factibles que resulta de un AFDM es tan grande que resulta imposible encontrar soluciones óptimas. Formular nuevas metaheurísticas que permitan reducir el espacio de soluciones factibles a un conjunto de soluciones óptimas, resulta ser una herramienta de gran importancia para una amplia gama de sectores a nivel productivo y significativa para la reducción de problemas ambientales a nivel mundial por el mal uso de los recursos no renovables.

Con el aumento de objetivos a optimizar en los procesos industriales, se hace necesario explorar métodos que permitan apuntar a cada vez más funciones objetivo. La formulación de un AFDM que permita la optimización tri-objetivo del problema del agente viajero simétrico multiobjetivo, es un herramienta excepcional para el sector industrial y constituye un avance significativo en la racionalización de recursos.

Se busca el diseño una herramienta que permita solucionar el problema planteado por medio del uso de un autómata finito determinista multi-objetivo que optimice problemas

del tipo tri-objetivo utilizando las técnicas de los algoritmos genéticos con lo que se busca evitar caer en óptimos locales.

Los pasos que siguen los algoritmos genéticos son:

- Generar la población inicial
- Calcular la función de aptitud para cada individuo de la población
- Repetir
 - Aplicar un operador de selección a los individuos de la población
 - Aplicar el operador genético de recombinación a los individuos de la población
 - Aplicar el operador genético de mutación a los individuos de la población
- Hasta que se cumpla una condición de terminación.

El modelo a diseñar se basa tanto en algoritmos genéticos como en autómatas finitos deterministas. Se plantea entonces la Metaheurística Basada En Autómatas Finitos Y Algoritmos Genéticos AGMIDA, que permita darle solución óptima al problema combinatorio del Agente Viajero Simétrico tri-objetivo.

- Primero crea un espacio de soluciones (estados) iniciales. Esto se logra asignándole pesos aleatorios a cada una de las funciones objetivo. Después, utiliza el método del vecino más próximo iniciando en diferentes ciudades y al finalizar almacena las soluciones obtenidas en forma de vector.
- Luego realiza un cruzamiento aplicando el operador genético de recombinación de la población. El cruzamiento se hace de la siguiente manera: Toma dos soluciones iniciales almacenadas y hace una partición aleatoria llamada k , para cruzar las soluciones en la k -ésima posición. Luego el algoritmo toma la cabeza del primero vector o padre y la combina con la cola del segundo vector o madre, como se

muestra en la figura 23. Como el problema del agente viajero simétrico no admite repeticiones, en caso de que al realizar el cruzamiento eso suceda, automáticamente el algoritmo llena los espacios del vector con otras ciudades que aún no hayan sido "visitadas" de manera aleatoria. El cruzamiento obtenido al aplicar el operador genético evita caer en óptimos locales.

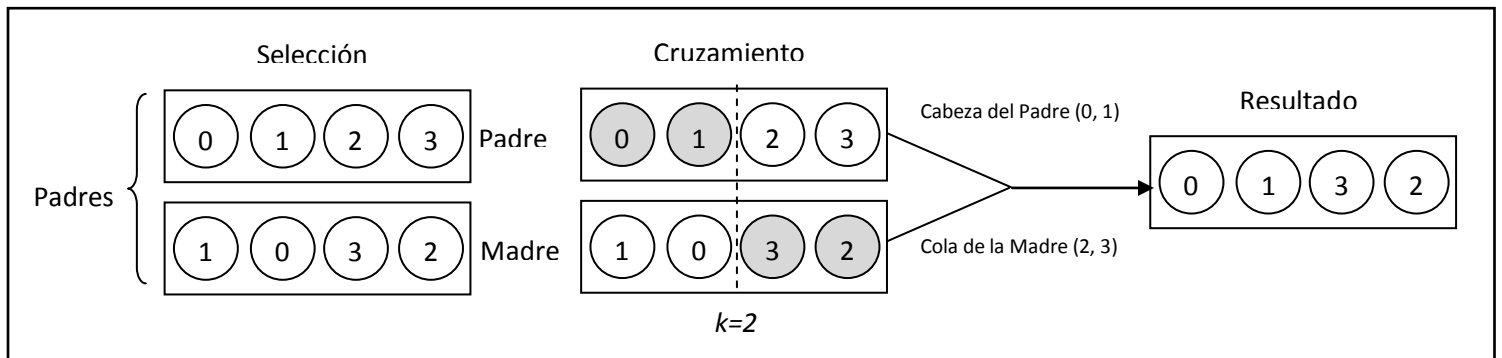


Figura 23- Cruzamiento Realizado en AGMIDA

Cuando se obtiene las soluciones iniciales con o sin mejoras, se recorrer el Autómata Finito Determinista Multi-Objetivo (AFDM), con el fin de buscar estados o soluciones no dominadas, de la siguiente manera:

1. Hacer Frente de Pareto Óptimo (FPO) como el conjunto de soluciones iniciales.
2. Para un número máximo de iteraciones establecidos haga:
3. Seleccione aleatoriamente un número de estados del conjunto .
4. Para cada estado seleccionado, aplique veces la función en busca de Soluciones No Dominadas en . Todos los estados encontrados cuyos no sean dominados por ningún se agregan a . Los elementos cuyos dominan por lo menos a un elemento se agregan al Conjunto de

Soluciones Elitista . Se seleccionan elementos en caso de que pasen iteraciones sin que una solución seleccionada en el paso 3 presente mejoras, teniendo en cuenta que .

5. Remove los elementos dominados de y . Ir a 2.

Se aproxima el conjunto de estados que contengan las soluciones óptimas del total de estados a partir de . En se almacenan los estados en donde las soluciones no son dominadas por soluciones correspondientes a estados pertenecientes a . contiene los estados con soluciones que en algún momento dominaron (y que no son dominadas) ó dominan a elementos de , por lo que proporciona la dirección a seguir para optimizar las soluciones que se encuentran en .

Formalmente, el framework de AGMIDA se muestra en la figura 24, y consta de tres métodos:

1. . Este método especifica los estados de que conformarán . Lógicamente es un conjunto implícito dado su gran tamaño (espacio de soluciones factibles). Por lo tanto, crear los estados que contienen las soluciones iniciales, equivaldrá a especificar los estados que conformarán que conformarán . Las soluciones iniciales pueden ser creadas de acuerdo al problema que se esta solucionando. Las soluciones obtenidas por técnicas heurísticas pueden ser una buena aproximación a las soluciones óptimas. Partir de un buen conjunto de soluciones iniciales puede ayudar a la técnica a obtener óptimos con menor esfuerzo computacional.
2. . Este método opcional permite mejorar las soluciones que fueron encontradas inicialmente. Esto se puede llevar a cabo

ya sea aplicando métricas a las soluciones obtenidas inicialmente ó replanteando el modelo original mediante problemas mono – objetivos para optimizar cada uno de los objetivos. Con esto se expande aún más el conocimiento de y con ello los posibles estados del AFDM donde las soluciones son óptimas.

3. . Esta última etapa de la metaheurística determina el modo de explorar el AFDM. De manera formal, este método puede ser apreciado en la figura 25. A continuación, se explican cada uno de los parámetros (variables) de entrada:

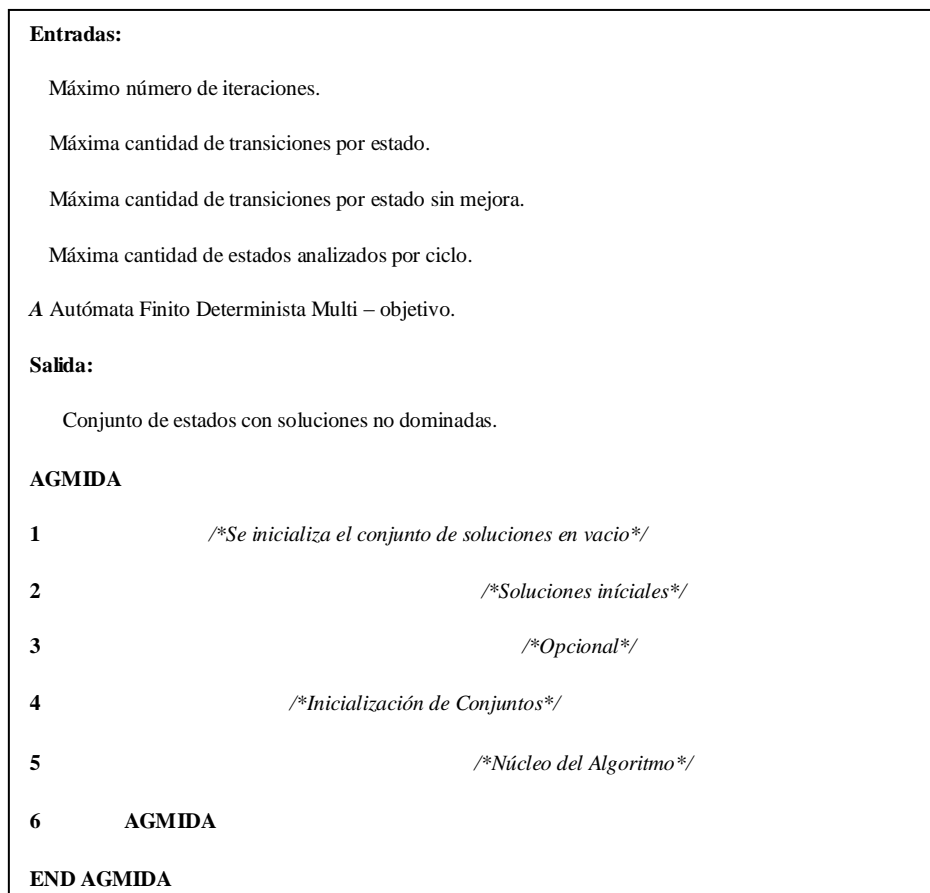


Figura 24- Framework de la Metaheurística AGMIDA

Donde S es un conjunto que contiene cada uno de los estados que conforman el conjunto solución del algoritmo. Al final, este conjunto contiene los estados cuyas soluciones conforman la aproximación al Frente de Pareto Óptimo. La condición con la cual se acepta un estado en este conjunto es que su solución no se encuentre dominada por alguna solución correspondiente a un estado perteneciente al conjunto. S es el conjunto que contiene los estados cuyas soluciones determinan la dirección del FPO. La condición necesaria para que un estado pertenezca a este conjunto es, además de su solución no ser dominada por ningún elemento de S , dominar por lo menos a una solución de un estado en S , $maxIter$ es un valor numérico entero que determina el máximo número de iteraciones de la metaheurística. n describe la cantidad de estados seleccionados aleatoriamente por iteración. $maxIterState$ es un valor numérico entero que indica el máximo número de iteraciones por cada estado seleccionado aleatoriamente. $maxIterAFDM$ Indica el máximo número de iteraciones permitidas por estado sin presentar mejoras. Es el AFDM que modela el problema combinatorio. Este especifica, los conjuntos S y S' , la función de transición T y las funciones objetivos f .

AGMIDA evita caer en óptimos locales al aplicarse el operador genético de recombinación de la población, es decir a realizar el cruzamiento mostrado en la figura 23. En la tabla 13 se muestran las variables utilizadas por la metaheurística AGMIDA en comparación los nombre utilizados tradicionalmente.

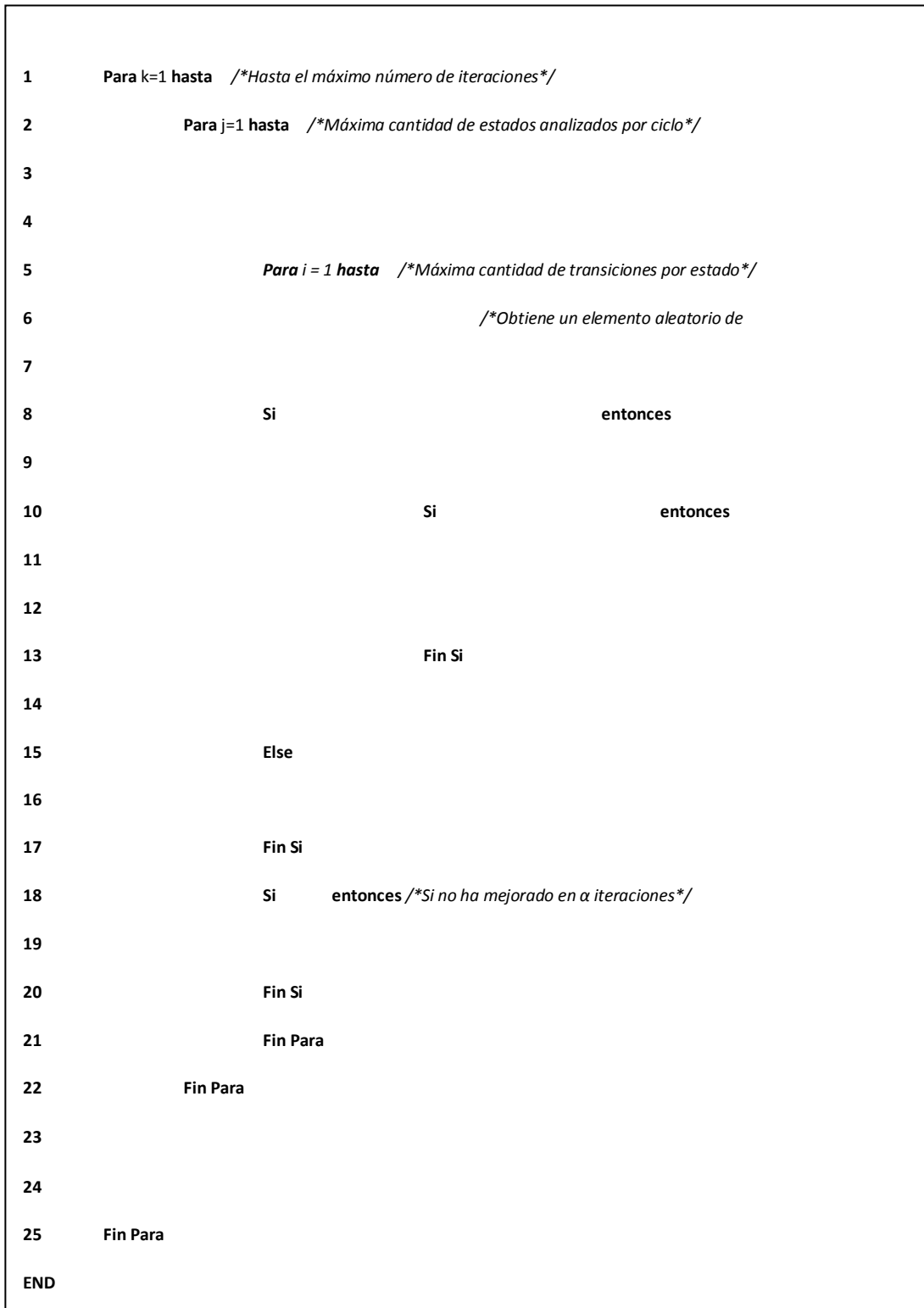


Figura 25- Framework del método

| Variable AGMIDA | Variable convencional | Significado |
|-----------------|-----------------------|--|
| | | Espacio de Soluciones Factibles |
| | | Soluciones Iniciales. |
| | | Frente de Pareto Óptimo. |
| | | Frente Elitista de Soluciones. |
| | Intercambio, Cambio | Mecanismo de Perturbación de Soluciones. |
| | | Máximo Número de Iteraciones |
| | | Máximo Número de Vecinos Analizados por Ciclo. |
| | | Funciones Objetivos del Problema Combinatorio. |

Tabla 13- Variables utilizadas en AGMIDA

3.2. ANALISIS DE LA COMPLEJIDAD DE AGMIDA

Asumamos valores muy grandes para n , m , y k :

(51)

Con n muy grande al igual que el resto de variables.

Analizaremos la complejidad se hará en el peor de los casos. En la figura del Framework de la Metaheurística AGMIDA, las líneas 1,4 y 6 suman 4 por las cuatro asignaciones realizadas. Las líneas 2,3 y 5 son sub-ecuaciones a calcular. Así pues, la ecuación inicial del análisis será:

(52)

Como n es un valor muy grande para todos los parámetros utilizados, las funciones f_1 y f_2 , asumiendo que crear un estado con su respectiva solución corresponden a una unidad de tiempo, equivalen a:

(53)

Resolviendo las sumatorias de 53 se obtiene:

(54)

En la figura que muestra un framework del método *explorar AFDM*, en el método , se inicia con un ciclo en la línea 1 pero el ciclo encierra otro. Al final del primer ciclo encontramos dos métodos que permiten remover soluciones dominadas de conjunto de estados – soluciones, como son n soluciones remover dominados implica contrastar cada solución contra el resto, asumiendo que comparar la solución de un estado con el de otro consume una unidad de tiempo, la ecuación inicial para es:

(55)

Al resolver doble sumatoria interna en 55 se obtiene:

(56)

La sumatoria interna de 56 corresponde al ciclo para de la línea 2 de la figura 25. Las líneas 3 y 4 suman dos, asumiendo que las asignaciones consumen una unidad de tiempo y teniendo en cuenta el ciclo para de la línea 5, la ecuación 56 se convierte en:

(57)

En la figura 25, las líneas 6, 7, 9, 11, 12 y 14 suman 6 por el supuesto realizado en los casos anteriores. Las líneas 8 y 10 demandan recorrer el conjunto de soluciones de tamaño n , sin embargo, con recorrerlo una sola vez es suficiente para obtener ambas informaciones, por lo tanto la ecuación 57 se convierte en:

(58)

Resolviendo las sumatorias de 58, se obtiene:

(59)

Reemplazando 54 y 59 en 52 se obtiene:

(60)

Organizando por coeficientes, la ecuación de tiempo en el peor de los casos para AGMIDA es:

(61)

Por lo tanto la cota superior de la metaheurística es igual a:

(62)

A partir de 61, se puede obtener la cota inferior de AGMIDA, el análisis es simple:

(63)

Por lo tanto de 63 se concluye que:

(63)

4. EXPERIMENTACION Y ANALISIS DE RESULTADOS

Para poder analizar de manera certera la efectividad de la metaheurística AGMIDA, se realizaron pruebas con instancias del Problema del Agente Viajero Simétrico (TSP), el cual es considerado un problema NP-hard y su planteamiento puede ser utilizado en gran variedad de problemas en la industria.

Se tomaron instancias de la librería TSPLIB⁶⁹, ya que son utilizadas por otros métodos de solución con los que se va a comparar los resultados de AGMIDA. Dicha librería contiene más de 200 de instancias del TSP y unas 500 instancias de problemas conocidos tales como el Ruteo de Vehículos el cual es un derivado del TSP. TSPLIB es administrada por el Grupo de Investigación de Optimización Discreta de la Universidad de Heidelberg en Alemania. Una gran cantidad de trabajos científicos relacionados con el tema utilizan estas instancias para probar la efectividad de sus métodos y compararlos con otros autores relacionados con el tema⁷⁰.

Tomando en cuenta lo anterior, el TSPLIB constituye una buena opción para probar y comparar los resultados obtenidos por AGMIDA con otros métodos a nivel mundial. Las instancias utilizadas de esta librería se encuentran en la tabla 14.

⁶⁹ TSPLIB. <http://comopt.ifi.uni-heidelberg.de/index.html>

⁷⁰ TSPLIB. <http://comopt.ifi.uni-heidelberg.de/publications/index.html>

| Nombre de la Instancia | Número de Ciudades | Número de Objetivos |
|------------------------|--------------------|---------------------|
| KROABC100 | 100 | 3 |
| KROBCD100 | 100 | 3 |

Tabla 14- Instancias de la TSPLIB utilizadas en las pruebas de AGMIDA

Los resultados obtenidos son contrastados contra otras técnicas utilizando las métricas presentadas en el marco teórico (Generación de Vectores no Dominados, Espaciamento, Distancia Generacional e Inversa de la Distancia Generacional).

4.1. ESPECIFICACIÓN DEL AFDM PARA LA PRUEBA

Para llevar a cabo la prueba primero es necesario definir el AFDM que permite modelar el Problema del Agente Viajero.

Siendo Q el espacio de soluciones factibles del TSP, cada estado s contiene un vector v_s que representa un camino para recorrer el conjunto de ciudades. Como el conjunto tiene un tamaño grande, este se encuentra de manera implícita en la memoria.

σ representa las 2 – tuplas que permiten intercambiar los elementos de un vector, para intercambiar las ciudades de una solución y obtener nuevas soluciones, es decir explorar nuevos estados de Q .

π es una función que toma una 2 – tupla de σ e intercambia de posición los elementos del vector solución de un estado. Esto es con el objetivo de alterar el orden de visita de las ciudades y así obtener mayor diversidad en las soluciones encontradas.

S_0 es el conjunto de soluciones iniciales. Las soluciones iniciales fueron creadas con la técnica heurística del vecino más cercano asignándoles pesos a las funciones. En la figura 26 se muestra el algoritmo del vecino más cercano o próximo.

| |
|---|
| <p><i>Inicialización</i> <i>Seleccionar un vértice j al azar.</i> <i>Hacer $t = j$ y $W = V \setminus \{j\}$.</i></p> <p><i>Mientras ($W \neq \emptyset$)</i> <i>Tomar j de $W / c_{ij} = \min \{c_{in} / i \text{ en } W\}$</i> <i>Conectar t a j</i> <i>Hacer $W = W \setminus \{j\}$ y $t = j$.</i></p> |
|---|

Figura 26- Algoritmo del Vecino Más Cercano

son las n matrices donde se encuentra las respectivas distancias entre cada una de las ciudades, siendo n el número de objetivos del problema. Las distancias son Euclidianas, se toma el orden de visita de las ciudades de t y se suman las respectivas distancias recorridas de la matriz.

4.2. COMPARACIÓN DE LOS RESULTADOS DE AGMIDA CON LAS METAHEURÍSTICAS SMOSA, CMOSA, UMOSA Y EMOSA

Los resultados obtenidos con AGMIDA, fueron contrastados con los resultados de otras metaheurísticas que resuelven el problema del agente viajero. Las técnicas con las que se hicieron las comparaciones son SMOSA, CMOSA, UMOSA y EMOSA.

Sr. Li, es el autor del EMOSA e implementador de los algoritmos UMOSA, CMOSA y SMOSA. Los algoritmos los corrió 20 veces cada uno y recolectó las soluciones no dominadas de todas las corridas. Para que la comparación sea en las mismas circunstancias, las soluciones no dominadas propuestas por AGMIDA fueron recolectadas de la misma forma.

Para comparar los resultados de las instancias KROABC100 y KROBCD100, por medio de las métricas GD e IGD, se tomaron los resultados obtenidos por H. Li y D. Landa-Silva en el artículo “*An Adaptive Evolutionary Multi-objective Approach Based on Simulated Annealing*”⁷¹.

⁷¹ Li, Hui; Landa-Silva, Dario. “An Adaptive Evolutionary Multi-objective Approach Based on Simulated Annealing”. Evolutionary Computation, ISSN 1063-6560 Marzo, 2011.

Los valores de los parámetros de AGMIDA para ejecutar la prueba son:

γ y β . Las figuras 27 y 28 ilustran los FP de cada algoritmo para cada una de las instancias. La tablas 15 y 16 muestran los valores para las métricas GVND, ESPACIAMIENTO, GD e IGD en cada una de las instancias.

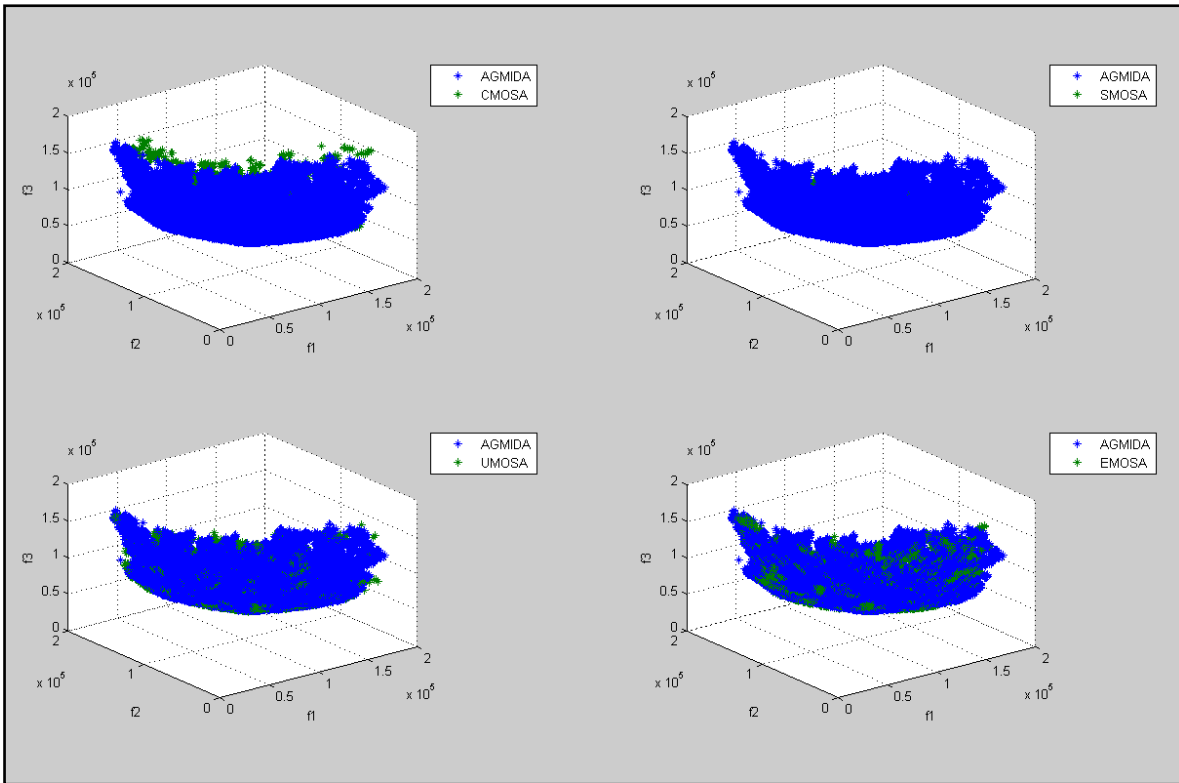


Figura 27- Contraste Visual del FP obtenido por AGMIDA en la instancia KROABC100, contra los FPs obtenidos por los algoritmos CMOSA, SMOSA, UMOSA y EMOSA

| | GVND | ESPACIAMIENTO | GD | IGD |
|---------------|-------|---------------|------------|------------|
| SMOSA | 1095 | 0,0599588 | 26,5312636 | 36172,0838 |
| CMOSA | 1817 | 0,04035959 | 14,3588074 | 29172,6591 |
| UMOSA | 2564 | 0,03498623 | 3,4396281 | 3333,40773 |
| EMOSA | 3194 | 0,03144919 | 2,38276567 | 2482,34369 |
| AGMIDA | 34077 | 0,01365555 | 0,05108237 | 129,865643 |

Tabla 15- Comparación de los resultados obtenidos por AGMIDA, EMOSA, CMOSA, SMOSA y UMOSA para la instancia KROABC100 utilizando las métricas GVND, ESPACIAMIENTO, GD e IGD

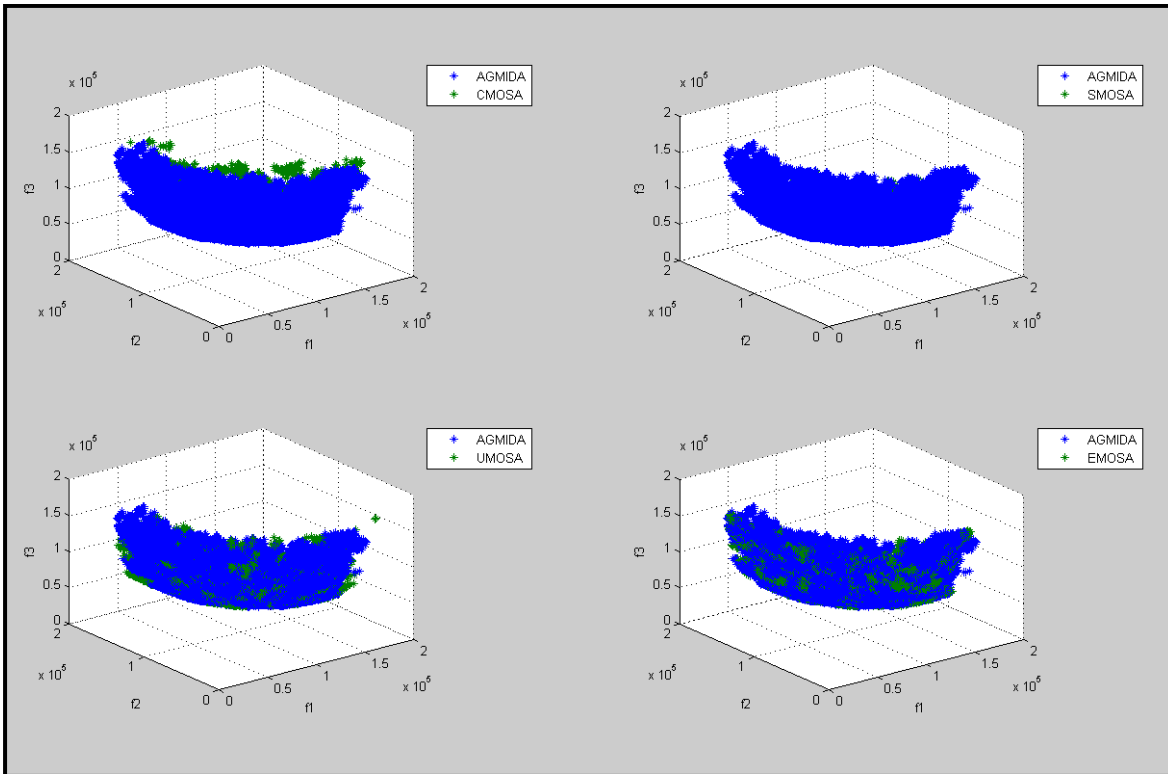


Figura 28- Contraste visual del FP obtenido por AGMIDA en la instancia KROBCD100, contra los FPs obtenidos por los algoritmos CMOSA, SMOSA, UMOSA y EMOSA

| | GVND | ESPACIAMIENTO | GD | IGD |
|---------------|--------------|-------------------|-------------------|-------------------|
| SMOSA | 1097 | 0,05749539 | 25,2829385 | 33752,4241 |
| CMOSA | 1795 | 0,04647346 | 14,1155524 | 28168,342 |
| UMOSA | 2472 | 0,03581236 | 3,46956584 | 3227,63056 |
| EMOSA | 3143 | 0,03150265 | 2,3144672 | 2321,81408 |
| AGMIDA | 34824 | 0,01307551 | 0,04979096 | 131,915227 |

Tabla 16- Comparación de los resultados obtenidos por AGMIDA, EMOSA, CMOSA, SMOSA y UMOSA para la instancia KROBDC100 utilizando las métricas GVND, ESPACIAMIENTO, GD e IGD

Se busca que GVND (Generación de Vectores no Dominados) sea un número grande, que el Espaciamiento sea un número pequeño al igual que la distancia generacional (GD) y la distancia generacional inversa (IGD).

Como puede apreciarse en las tablas 15 y 16, los resultados obtenidos con AGMIDA en tienen mejor desempeño con las métricas GVND, ESPACIAMIENTO, GD e IGD que SMOSA, CMOSA, UMOSA y EMOSA, mostrando una superioridad notable en los resultados para las dos instancias KROABC100 y KROBCD100. En la tabla 17 se puede observar el promedio de los resultados obtenidos.

| | GVND | SPACING | GD | IGD3 |
|---------------|-------------|----------------|------------|-------------|
| SMOSA | 1096 | 0,0587271 | 25,9071011 | 34962,2539 |
| CMOSA | 1806 | 0,04341652 | 14,2371799 | 28670,5005 |
| UMOSA | 2518 | 0,0353993 | 3,45459697 | 3280,51915 |
| EMOSA | 3168,5 | 0,03147592 | 2,34861643 | 2402,07889 |
| AGMIDA | 34450,5 | 0,01336553 | 0,05043667 | 130,890435 |

Tabla 17- Promedios de los resultados obtenidos por AGMIDA, EMOSA, CMOSA, SMOSA y UMOSA para las instancias KROABC100 y KROBDC100 utilizando las métricas GVND, ESPACIAMIENTO, GD e IGD

5. CONCLUSIONES Y TRABAJOS FUTUROS

Las conclusiones que resultan del trabajo de investigación presentado son citadas a continuación.

1. Se diseño e implemento una metaheurística basada en Autómatas Finitos Deterministas Multi – Objetivo y algoritmos genéticos (AGMIDA) que permite dar solución al problema del agente viajero simétrico tri-objetivo. Al basarse en la teoría de AFDM, las soluciones obtenidas con el método no son del tipo infactible.
2. Al contrastar los resultados con las técnicas SMOSA, CMOSA, UMOSA y EMOSA, con tres objetivos usando las instancias KROABC100 y KROBDC100, AGMIDA logra superar en todas las métricas a las otras técnicas. Todos los resultados comparados contra AGMIDA pertenecen a escritos ISI.
3. La implementación AGMIDA logra ser una contribución importante para la industria, dado que mejora los resultados obtenidos por otras técnicas. El uso de esta técnica podría constituir un ahorro significativo en materiales y uso racional de recursos naturales no renovables.

Los trabajos que se proponen a futuro son:

1. Formular una mejora para AGMIDA que permita resolver otros problemas combinatorios multi – objetivo tales como el ruteo de vehículos, el problema de la mochila y la p – mediana.

2. Escribir un artículo ISI con las pruebas realizadas al resolver las instancias del Problema del Agente Viajero.
3. Contrastar los tiempos obtenidos por AGMIDA contra los consumidos por otras técnicas.
4. Formular AGMIDA para una mayor cantidad de funciones objetivos.

6. REFERENCIAS BIBLIOGRAFICAS

- Correa Posada, Carlos Mario. "Modelo de Optimización para las Plantas Térmicas de Generación de Ciclo Combinado en el Despacho Económico", Universidad nacional de Colombia. Medellín. 2009.
- Peiyou Chen; Xinming Xu. "A Hybrid Algorithm for Multi-Depot Vehicle Routing Problem", Service Operations and Logistics, and Informatics, IEEE/SOLI 2008, IEEE International Conference, vol. 2, pag. 2031, 25 Noviembre, 2008.
- Lastres Aleaga, Arlys Michel; Cordovés Garcias, Alexis; Arzola Ruis, José. "Optimización de la Distribución y Corte de Piezas Irregulares en Chapas para la Industria Minera", Minería y Geología, vol.26, no.3, pag. 68-80, Julio-Septiembre, 2010.
- Niño, Elías; Ardila, Carlos. "Algoritmo Basado en Autómatas Finitos Deterministas para la obtención de óptimos globales en problemas de naturaleza combinatoria", Revista de Ingeniería y Desarrollo, No 25, pag. 100 – 114. ISSN 0122 – 3461, 2009.
- Dorigo, Marco; Maniezzo, Vittorio; Colorni, Alberto. "The Ant System: Optimization by a Colony of Cooperating Agents" Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions, vol.26, no.1, pag. 29-41, Febrero, 1996
- Iredi, Steffen ; Merkle, Daniel; Middendorf, Martin. "Bi-criterion optimization with multi colony ant algorithms", First International Conference on Evolutionary Multi-criterion Optimization, Lecture Notes in Computer Science, pag 359–372, 2001
- Mariano, Carlos; Morales, Eduardo. "A multiple objective Ant-Q algorithm for the design of water distribution irrigation networks", Technical Report HC-9904, Instituto Mexicano de Tecnología del Agua, Mexico, Junio, 1999
- Gambardella, Luca; Taillard, Éric; Agazzi, Giovanni. "MACS-VRPTW: A Multiple Colony System For Vehicle Routing Problems With Time Windows", New Ideas in Optimization, McGraw-Hill, 1999, pag.. 73–76, 1999.
- Doerner, Karl; Gutjahr, Walter; Hartl, Richard; Strauss, Christine; Stummer, Christian. "Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection", Annals of Operations, pag. 79–99, 2004

- Cardoso, Pedro; Jesus, Mario ; Marquez, Alberto .“MONACO- Multi-Objective Network Optimization based on an ACO”, Proc. X Encuentros de Geometría Computacional, Sevilla, Spain, Junio 16–17, 2003.
- Doerner, Karl; Hartl, Richard; Reimann, Marc. “Are COMPETants more competent for problem solving?—The Case of Full Truckload Transportation”, Central European Journal of Operations Research, pag. 115–141, 2003
- Tse Guan Tan; Hui Keng Lau; Teo, Jason. "Cooperative Coevolution for Pareto Multiobjective Optimization: An Empirical Study Using SPEA2", TENCON 2007 - 2007 IEEE Region 10 Conference, pag.1-4, Octubre 30 -Noviembre 2, 2007.
- Czyzak, Piotr; Jaskiewicz, Adrezej. “Pareto simulated annealing - a metaheuristic technique for multiobjective combinatorial optimization”, Journal of Multi-Criteria Decision Analysis, vol. 7, no. 1, pag. 34–37, Diciembre, 1998.
- Ulungu; Teghem; Fortemps; Tuytens, “MOSA method: A tool for solving multiobjective combinatorial optimization problems,” Journal of Multi-Criteria Decision Analysis, vol. 8, no. 4, pp. 221–236, 1999.
- Bandyopadhyay, Sanghamitra; Saha, Sriparna; Maulik, Ujjiwal; Deb, Kalyanmoy. "A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA”, Evolutionary Computation, IEEE Transactions, vol.12, no.3, pag.269-283, Junio, 2008
- Borbulya, István. “An EC-Memory based Method for the Multi-Objective TSP”, Proceedings of the 9th annual conference on Genetic and evolutionary computation, New York, 2007.
- Borgulya, István. “An Evolutionary Algorithm for the bi-objective QAP”, Computational Intelligence, Theory and Applications Advances, springer series, part 22, pag. 577-586, 2006.
- Niño, Elías; Ardila, Carlos; Jabba, Daladier; Donoso, Yesid. “A Novel Algorithm Base On Deterministic Finite Automaton for Solving the Mono-Objective Symmetric Traveling Salesman Problem”, International Journal of Artificial Intelligence, vol. 5, No A10. pag. 510, North America, 2010.
- Cagnina, Leticia. “Optimización Mono y Multiobjetivo a través de una Heurística de Inteligencia Colectiva”, Facultad de Ciencias Físico Matemáticas y Naturales, Universidad Nacional de San Luis, Argentina, 2010.
- Donoso, Yezid; Fabregat, Ramón. “Multi-Objective Optimization in Computer Networks Using Metaheuristics”, pag. 1 – 3,Aurbach Publications, New York, United States 2007.
- Figuera, Pau. “Optimización de productos y procesos industriales”, cap. 1, Ediciones Gestión 2000, Barcelona, España, 2006.

- Tovar, Luis; Coronael, Margarita; Donoso, Yesid. “Multi Optimización Multiobjetivo en Redes Ópticas con Transmisión Multicast, Utilizando Algoritmos Evolutivos y Lógica Difusa”, Ingeniería & Desarrollo, Número 21, Enero-Junio 2007.
- Cruz, Nareli; Coello, Carlos. “Un Sistema Inmune Artificial para Solucionar Problemas de Optimización Multiobjetivo”, Grupo de Computación Evolutiva Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2003.
- López, Antonio. “Diseño de un Algoritmo Evolutivo Multiobjetivo Paralelo”, Grupo de Computación Evolutiva Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2005.
- Quezada, Fátima; Padilla, Felipe; De León, Eunice; Caballos, Rebeca. “Sistema Inmune Artificial para Resolver el Problema del Agente Viajero”, Segundo Congreso Estatal “La Investigación en el Posgrado”, México, Noviembre, 2010
- Castro, Salvador. “Creación de Portafolios de Inversión Utilizando Algoritmos Evolutivos Multiobjetivo”, Grupo de Computación Evolutiva Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2005.
- Glover, Fred. “Future Paths for Integer Programming and Links to Artificial Intelligence”, Computers and Operations Research, vol. 13, No. 5, pag. 533-349, 1986.
- Díaz, Adenso. “Recocido Simulado”, Universidad de Oviedo, 2004.
- Kirkpatrick, S.; Gellatt, C; Vecchi, M. “Optimization by Simulated Annealing”, Science, 220: pag. 671-680, 1983.
- Cerny, V. “Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm”, Journal of Optimization Theory and Applications, pag. 41—51, 1985.
- Dorigo, M.; Maniezzo, V.; Colorni, A. “The Ant System: Optimization by a Colony of Cooperating Agents”, IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1):29-41, 1996.
- Mendoza, Benito. “Uso del Sistema de la Colonia de Hormigas para Optimizar Circuitos Lógicos Combinatorios”, Universidad Veracruzana, México, 2001.
- Glover, Fred. “Future Paths for Integer Programming and Links to Artificial Intelligence”, Computers and Operations Research, vol. 13, No. 5, pag. 533-349, 1986.
- Melián, Belén; Glover, Fred. “Introducción a la Búsqueda Tabú”. Dpto. de Estadística, I.O. y Computación, Escuela Técnica Superior de Ingeniería Informática, Universidad de La Laguna.

- Tupia, Manuel. "Un Algoritmo GRASP para Resolver el Problema de la Programación de Tareas Dependientes en Máquinas Diferentes (Task Scheduling)", Universidad Nacional Mayor de San Marcos, Lima, Perú, 2005.
- Castro, David. "Teoría De Autómatas y Lenguajes Formales y Gramática", Universidad de Alcalá, 2003.
- Niño, Elías; Ardila, Carlos , “Algoritmo Basado en Autómatas Finitos Deterministas para la Obtención de Óptimos Globales en Problemas de Naturaleza Combinatoria”, Revista de Ingeniería y Desarrollo, No 25. pp 100 – 114, ISSN 0122 - 3461.
- Niño, Elías, "Algoritmo Basado en Autómatas Finitos Deterministas Y Expresiones Regulares para solucionar el Problema del Agente Viajero Simétrico Multi-Objetivo ", VIII Congreso de Investigación Operativa - Optima 2009, Chile, 2009.
- Niño, Elías; Ardila, Carlos; Jabba, Daladier; Donoso, Yesid, "Mods: A Novel Metaheuristic Of Deterministic Swapping For The Multi – Objective Optimization Of Combinatorial Problems ".
- TSPLIB. <http://comopt.ifl.uni-heidelberg.de/index.html>
- TSPLIB. <http://comopt.ifl.uni-heidelberg.de/publications/index.html>
- Li, Hui; Landa-Silva, Dario. “An Adaptive Evolutionary Multi-objective Approach Based on Simulated Annealing”. Evolutionary Computation, ISSN 1063-6560 Marzo, 2011.