

**ENFOQUE MULTIOBJETIVO DE UNA METAHEURÍSTICA PARA LA  
GENERACIÓN Y OPTIMIZACIÓN DE PLANES DE RUTEO PARA EL  
TRANSPORTE TERRESTRE DE PASAJEROS EMPLEANDO LÍNEAS  
PREDEFINIDAS, BASADA EN TÉCNICAS DE ALGORITMOS EVOLUTIVOS  
HÍBRIDOS**

**JULIAN LOPEZ FRANCO**



**FUNDACIÓN UNIVERSIDAD DEL NORTE  
DIVISIÓN DE INGENIERÍAS  
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL  
BARRANQUILLA, AGOSTO DE 2010**

**ENFOQUE MULTIOBJETIVO DE UNA METAHEURÍSTICA PARA LA  
GENERACIÓN Y OPTIMIZACIÓN DE PLANES DE RUTEO PARA EL  
TRANSPORTE TERRESTRE DE PASAJEROS EMPLEANDO LÍNEAS  
PREDEFINIDAS, BASADA EN TÉCNICAS DE ALGORITMOS EVOLUTIVOS  
HÍBRIDOS**

**DIRECTORES**

**Ph.D. CARLOS D. PATERNINA**

**M.Sc. IVÁN SAAVEDRA A.**



Proyecto de investigación y desarrollo realizado como requisito para optar al título  
de Maestría en Ingeniería Industrial

**FUNDACIÓN UNIVERSIDAD DEL NORTE  
DIVISIÓN DE INGENIERÍAS  
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL  
BARRANQUILLA, AGOSTO DE 2010**

**NOTA DE APROBACIÓN**

**VISTO BUENO DEL DIRECTOR**

---

Ph.D. Carlos D. Paternina  
Ingeniero Industrial

---

M.Sc. Iván Saavedra A.  
Ingeniero de Sistemas

## **AGRADECIMIENTOS**

Como primera medida agradezco a Dios y a mi señor Jesucristo por la oportunidad y las grandes bendiciones que me han dado a través de este proceso de maestría, hacia mi desarrollo personal y profesional. Doy gracias a Dios pues El me devolvió al camino que había perdido y me permitió retomarlo con gran optimismo y bendición.

A mi esposa, Brooke, por su apoyo incondicional, el ánimo y amor que me expreso durante todo este tiempo, fueron fundamentales para no retroceder en mi decisión de terminar este trabajo. A mi familia, padres y hermanos, que día a día con sus oraciones, buenos deseos y apoyo hicieron posible este logro.

A mis tutores y amigos, Iván Saavedra y Carlos Paternina por su valioso tiempo, apoyo y consejos profesionales en pos de la consecución de este trabajo de tesis de maestría.

Gracias a mis amigos y compañeros que de una u otra forma estuvieron dándome aliento y apoyo en este proceso. Sin ustedes no se podría haber conseguido este logro. Dios los bendiga.

## RESUMEN

En el mundo real existe una gran cantidad de problemas de optimización para los cuales los métodos de programación matemática no pueden garantizar que la solución obtenida sea óptima y, en muchos de ellos, ni siquiera pueden aplicarse. En estos problemas las metaheurísticas se vuelven una alternativa viable. De entre las diversas metaheurísticas disponibles en la actualidad, los algoritmos Genéticos Híbridos se cuentan entre los más populares debido a su simplicidad conceptual y su eficacia.

Hoy en día, los problemas de optimización multiobjetivo se presentan con mucha frecuencia. En ellos hay que optimizar dos o más funciones objetivo que se encuentran normalmente en conflicto entre sí. La solución de problemas multiobjetivo usando metaheurísticas (sobre todo, Algoritmos Genéticos Híbridos) es un tema que ha adquirido gran popularidad en los últimos tiempos.

En este documento presentamos un Algoritmo Genético Híbrido Multiobjetivo para la generación y optimización de planes de ruteo, asignación de frecuencias y horarios de despacho empleando líneas predefinidas. Además se documenta un caso de real de uso en una empresa transportadora de pasajeros terrestre en Colombia.

## TABLA DE CONTENIDO

<b>AGRADECIMIENTOS</b> .....	<b>4</b>
<b>RESUMEN</b> .....	<b>5</b>
<b>TABLA DE CONTENIDO</b> .....	<b>6</b>
<b>LISTA DE FIGURAS</b> .....	<b>8</b>
<b>GLOSARIO</b> .....	<b>11</b>
<b>1. INTRODUCCIÓN</b> .....	<b>12</b>
1.1 <i>INTRODUCCIÓN AL PROBLEMA</i> .....	12
1.2 <i>PLANTEAMIENTO DEL PROBLEMA</i> .....	14
1.3 <i>OBJETIVOS</i> .....	15
1.4 <i>ALCANCES Y LIMITACIONES</i> .....	16
<b>2. CONCEPTOS, TÉCNICAS Y ESTRATEGIAS DE OPTIMIZACIÓN MULTIOBJETIVO</b> .....	<b>18</b>
2.1 <i>DESCRIPCIÓN</i> .....	18
2.2 <i>CONCEPTOS Y BASES TEÓRICAS</i> .....	18
2.3 <i>TÉCNICAS METAHEURÍSTICAS</i> .....	19
<b>3. FORMULACION MATEMÁTICA DEL PROBLEMA</b> .....	<b>50</b>
3.1 <i>DESCRIPCIÓN</i> .....	50
3.2 <i>DEFINICIÓN DE ÍNDICES Y CONJUNTOS</i> .....	50
3.3 <i>DEFINICIÓN DE PARAMETROS Y VARIABLES</i> .....	51
3.4 <i>FUNCIONES OBJETIVO</i> .....	53

3.5 RESTRICCIONES .....	54
<b>4. SELECCIÓN, DISEÑO Y DESCRIPCIÓN DE UNA METAHEURÍSTICA PARA LA GENERACIÓN Y OPTIMIZACIÓN MULTI OBJETIVO DE PLANES DE RUTEO PARA EL TRANSPORTE TERRESTRES DE PASAJEROS.....</b>	<b>57</b>
4.1 DESCRIPCIÓN .....	57
4.2 COMPARACIÓN Y SELECCIÓN DE TÉCNICAS PARA LA SOLUCIÓN DEL PROBLEMA PLANTEADO.....	57
4.3 ALGORITMOS GENÉTICOS HÍBRIDOS MULTI OBJETIVO .....	61
<b>5. CASO DE APLICACIÓN: DSS-BRASILIA SISTEMA DE SOPORTE DE DECISIONES EN TRANSPORTE TERRESTRE DE PASAJEROS. ....</b>	<b>88</b>
5.1 DESCRIPCIÓN .....	88
5.2 EXPLICACIÓN PROBLEMA A RESOLVER.....	88
5.3 RESTRICCIONES, FUNCIONES Y VARIABLES ESPECÍFICAS DEL PROBLEMA.....	90
5.3 EVALUACIÓN Y RESULTADOS.....	98
<b>6. CONCLUSIONES Y DESARROLLOS FUTUROS.....</b>	<b>102</b>
6.1 CONCLUSIONES .....	102
6.2 DESARROLLOS Y TRABAJOS FUTUROS .....	103
<b>BIBLIOGRAFIA.....</b>	<b>106</b>

## LISTA DE FIGURAS

Figura 1. Estructura básica de una línea .....	14
Figura 2. Optimo global y óptimo local .....	23
Figura 3. Proceso para memoria de Corto plazo en búsqueda Tabú. ....	25
Figura 4. Proceso de selección de mejor movimiento posible en tabú.....	26
Figura 5. Estructura de operación del Simulated Annealing.....	27
Figura 6. Comportamiento de colonias de hormigas naturales .....	30
Figura 7. Estructura del proceso de la búsqueda dispersa .....	31
Figura 8. Ejemplo de la optimalidad Pareto en el espacio objetivo.....	34
Figura 9. Estructura algorítmica básica del SPEA .....	38
Figura 10. Estructura Algorítmica básica de un MOTS.....	39
Figura 11. Estructura algorítmica de un MOSA.....	41
Figura 12. Estructura básica de funcionamiento de MOGRASP .....	43
Figura 13. Estructura algorítmica básica de un MOACS .....	44
Figura 14. Estructura algorítmica de un MOSS .....	46
Figura 15. Algoritmo MOGRASP para el problema TNDP .....	48
Figura 16. Estructura del cuerpo del plan de ruteo con n niveles de servicio .....	58
Figura 17. Comportamiento de algoritmos genéticos híbridos (AGH) .....	62
Figura 18. Estructura básica de AGHMO planteado .....	63
Figura 19. Estructura completa del cromosoma para una solución factible .....	64
Figura 20. Estructura de heurística para generación de planes semillas. ....	66
Figura 21. Función para generar cuerpo del plan de ruteo .....	67
Figura 22. Función de planificación de líneas predefinidas para el cuerpo generado .....	68

Figura 23. Rutina de descomposición de líneas.....	68
Figura 24. Ejemplo de descomposición de líneas .....	69
Figura 25. Rutina general para asignación de horarios de despacho.....	70
Figura 26. Función para evaluar número de despachos .....	71
Figura 27. Evaluación de cruces .....	72
Figura 28. Ciclo del AGHMO .....	72
Figura 29. Función para cálculo de funciones .....	73
Figura 30. Evaluación de costos anticipos.....	74
Figura 31. Función de evaluación de ventas .....	74
Figura 32. Función de no-dominancia.....	76
Figura 33. Función de reducción aplicando clustering.....	77
Figura 34. Cálculo del Strenght y Fitness de población No-Dominada .....	78
Figura 35. Calculo de fitness de población dominada .....	78
Figura 36. Selección utilizando el método de la ruleta.....	79
Figura 37. Estructura del cruzamiento general para el AGHMO .....	80
Figura 38. Función para el operador de cruzamientos .....	81
Figura 39. Estructura del operador de mutación general.....	82
Figura 40. Función para el operador de mutación .....	82
Figura 41. Estructura de funcionamiento del optimizador local.....	83
Figura 42. Función de optimización local.....	84
Figura 43. Variaciones en el espacio de búsqueda de un AGHMO.....	85
Figura 44. Mapa de operaciones de Expreso Brasilia S.A. ....	89
Figura 45. Estructura de matriz origen, destino, alterna .....	91
Figura 46. Parámetros de operación del AGHMO desde el DSS-Brasilia.....	93
Figura 47. Función para eliminación de cruces entre agencias DSS-Brasilia.....	95
Figura 48. Funcionamiento del operador de cruzamiento a nivel de la línea. ....	95
Figura 49. Resultado del cruce de líneas .....	96
Figura 50. Planes hijos producto del cruce entre líneas .....	97
Figura 51. Selección de una línea y punto de corte para mutación .....	97
Figura 52. Tabla de comparativa de resultados del AGHMO vs Operacion Real ..	99

Figura 53. Frente Pareto descrito por el AGHMO .....	99
Figura 54. Reporte del DSS-Brasilia aplicando el AGHMO .....	100
Figura 55. Operadores genéticos del EMMRS .....	104

## **GLOSARIO**

SOP	Single Objective Problem
MOP	Multi Objective Problem
AG	Algoritmo Genético
AGH	Algoritmo Genético Híbrido
AGHMO	Algoritmo Genético Híbrido Multiobjetivo
AE	Algoritmo Evolutivo
SS	Scatter Search
SA	Simulated Annealing
TS	Tabu Search
ACS	Ant Colony Systems
OCH	Optimización de Colonia de Hormigas
GRASP	Greedy Randomized Adaptive Search Procedure
MOEA	MultiObjective Evolutionary Algorithm
MOACS	MultiObjective Ant Colony Systems
MOSS	Multiobjective Scatter Search
MOTS	Multiobjective Tabu Search
MOSA	Multiobjective Simulated Annealing
MOGRASP	Multiobjective GRASP
DSS	Decision Support System
PFtrue	Frente de Pareto Verdadero
FP	Frente de Pareto.
TNDP	Transit Network Design Problem

## **1. INTRODUCCIÓN**

Este capítulo describe todos los aspectos que motivaron y dieron pie al desarrollo de este trabajo de tesis, empezando por el planteamiento del problema específico referente a la óptima planeación de líneas de transporte terrestre de pasajeros utilizando rutas y líneas cíclicas predefinidas tomando en cuenta múltiples objetivos. El capítulo finaliza con los objetivos y alcances del trabajo.

### **1.1 INTRODUCCIÓN AL PROBLEMA**

Hoy en día, las empresas de transporte de pasajeros necesitan buscar nuevas alternativas para aumentar sus ingresos, mejorar la calidad del servicio y reducir sus costos operativos. Por lo tanto, la optimización de un sistema de transporte de pasajeros plantea múltiples objetivos del tipo: minimizar tiempos de viaje, maximizar la calidad del servicio y maximizar el beneficio de las empresas transportistas. Estos objetivos son generalmente contrapuestos, es decir, una mejora en uno implica un empeoramiento en el otro.

Eso se puede ver con claridad por ejemplo, en empresas de transporte que cuentan con una memoria histórica de datos, planes de ruteo, líneas, frecuencias, horarios de despacho, asignaciones de vehículos etc., la cual ha permitido por algún tiempo cubrir la demanda real en diferentes épocas o temporadas de operación para sus diferentes destinos. Esto convierte a este caso específico, en una operación de optimización mono-objetivo, sin que necesariamente un

mejoramiento en la cobertura de su demanda real o proyectada, implique una minimización de costos operativos.

*“La planificación del transporte de pasajeros basada en herramientas de apoyo a la decisión cobra cada vez más importancia, tanto en los países desarrollados como en los en vías de desarrollo”<sup>1</sup>.*

Es importante notar que a nivel algorítmico y matemático, el problema de optimización de rutas, frecuencias y asignación de flota, posee varias fuentes de complejidad (no linealidad, no convexidad, múltiples objetivos) que dificultan tanto su formulación como la derivación de algoritmos eficientes de resolución. Este tipo problemas son resueltos a través de técnicas combinatorias o algoritmos aproximados, en donde la solución es un subconjunto de un conjunto total de soluciones, por lo cual pertenece a la clase de los NP-HARD o NP-Complejos, es decir, que se pueden plantear soluciones que se acerquen a una óptima en un tiempo polinomial con un consumo razonable de recursos, lo que ofrece una alternativa muy atractiva a la búsqueda determinística o exacta de la solución óptima.

Entre los algoritmos aproximados, se encuentran los basados en métodos constructivos y otros de búsqueda local. En los últimos años, a los métodos básicos de resolución se han agregado técnicas más generales llamadas Metaheurísticas, que son técnicas combinatorias o estrategias de alto nivel que guía a otras heurísticas para buscar soluciones factibles en dominios donde la tarea es compleja y a su vez, permiten optimizar los recursos y minimizar los costos asociados al liberar recursos.

Este trabajo de grado plantea el diseño y desarrollo de una metaheurística que permita la creación, evaluación, combinación y optimización multiobjetivo de

---

<sup>1</sup> Begoña G, Moreno J diseño robusto de rutas y frecuencias en el transporte público urbano.

planes y líneas de ruteo predefinidas utilizando algoritmos genéticos híbridos o de búsqueda genética local.

## 1.2 PLANTEAMIENTO DEL PROBLEMA

Algunas empresas del sector de transporte de pasajeros no cuentan con la posibilidad de realizar pruebas o combinaciones de sus rutas origen – destino con libertad, dadas muchas restricciones a las que están sujetas por los gobiernos como primera instancia. Luego se encuentran razones como la demanda de ciertas agencias y la competencia que les causan otras empresas.

Por tales razones, estas empresas han desarrollado un conjunto de recorridos, los cuales cumplen con las restricciones de la empresa y del gobierno. Algunas de ellas con herramientas que soporten y complementen el conocimiento y la experiencia profesional. Estas líneas definidas, están compuestas por un gran número de rutas o-d, donde se cumple que el origen inicial de la línea es el destino final de la misma, como está ilustrado a continuación.

**Figura 1. Estructura básica de una línea**



Fuente:Propia

En esta se indica que el primer tramo de la línea es desde Agencia inicial hasta Agencia B, el siguiente desde Agencia B hasta Agencia C, así sucesivamente hasta que regresan a Agencia inicial. El reto más grande para estas empresas, es saber de qué forma construir un plan de recorridos para los diferentes niveles de servicio que poseen utilizando estas líneas que han definido, pero que satisfaga varios objetivos planteados, como son la minimización del costo del plan, la maximización de las ventas por cada plan y la mayor satisfacción de la demanda para las agencias operativas. Por otro lado, las herramientas que presentan

algunas aproximaciones, plantearon soluciones a corto o mediano plazo, pero es conocido que los cambios de la operación en cualquier momento pueden tener un costo importante de implantación financiero, político y social, debiéndose por ello, considerar también el largo plazo.

Este trabajo plantea una metaheurística para optimización multiobjetivo, construcción y evaluación de planes de recorrido, frecuencias de salidas, horarios y eliminación de cruzamientos, utilizando líneas predefinidas. Las funciones objetivo que se deberán optimizar son la minimización del costo anticipado y la maximización de las ventas para cada plan de recorrido diseñado. La metaheurística estará basada en técnicas de algoritmos evolutivos híbridos que permiten la optimización global tanto como local, permitiendo en un tiempo polinomial converger a soluciones operativas muy cercanas a la óptima.

Los beneficios que tiene realizar este tipo de trabajos se relacionan con hacer que los recursos que se tienen sean utilizados de la forma más eficiente posible, y que los costos asociados se minimicen con el fin de redistribuir los recursos liberados en pos de mejorar la situación existente.

### **1.3 OBJETIVOS**

#### **1.3.1 OBJETIVO GENERAL**

Diseñar y desarrollar una Metaheurística para la generación y optimización multiobjetivo de planes de ruteo para el transporte terrestre de pasajeros empleando líneas predefinidas, basada en técnicas de algoritmos evolutivos híbridos.

#### **1.3.2 OBJETIVOS ESPECIFICOS**

- Estudiar, identificar y comparar los mejores modelos metaheurísticos multiobjetivo a seguir para la solución del problema de transporte de pasajeros planteado.

- Desarrollar una heurística para generación de planes semillas tomando como objetivo la minimización de costos anticipados.
- Diseñar un algoritmo que gestione la asignación de los mejores horarios y frecuencias de salidas de acuerdo a la demanda acumulada para los diferentes o-d.
- Diseñar una técnica no exhaustiva para la eliminación de los cruzamientos que se presentan entre rutas y líneas que componen un plan de ruteo.
- Hacer pruebas en un sistema de soporte de decisiones para una empresa del sector de transporte de pasajeros.

#### **1.4 ALCANCES Y LIMITACIONES**

El alcance del proyecto se encuentra limitado solo a la modelación y construcción de una metaheurística que genere planes de ruteo basados en líneas y rutas predefinidas, y realice optimización multiobjetivo sobre los planes diseñados. Se tendrá en cuenta para este desarrollo que:

- Los únicos objetivos que evaluará la metaheurística y optimizará serán, la minimización de costos anticipados para cada plan generado y la maximización de las ventas de cada plan.
- Los valores de demanda son entregados como parámetros. Ni la metaheurística, ni ningún algoritmo empleado en ella genera pronósticos de demanda. Estos datos son entradas del sistema o del algoritmo.
- La planificación de líneas definidas se hará por los diferentes niveles de servicio, de acuerdo a la cantidad de buses disponibles para cada nivel y número de días que debe abarcar el plan.
- Ninguna heurística empleada en el proyecto, asignará flota ni conductores a los planes de ruteo diseñados. Esto hace parte de otro trabajo.
- La planificación y eliminación de cruzamientos se hará sin una búsqueda exhaustiva para permitir que la razón de ser de un método de optimización metaheurístico tenga cabida.

- Los mejores horarios de despachos serán entregados como entrada del algoritmo entre todos los O-D permitidos.
- Se escogerá solo un método metaheurístico para modelar el algoritmo planteado, sin la necesidad de programarlos para escoger el de mejores resultados.

## **2. CONCEPTOS, TÉCNICAS Y ESTRATEGIAS DE OPTIMIZACIÓN MULTIOBJETIVO**

### **2.1 DESCRIPCIÓN**

En este capítulo, en su primera parte, se describe toda la base de optimización aproximada o estocástica, fundamentada en metaheurísticas. Al final del capítulo se muestran algunas de sus aplicaciones a los problemas de transporte, haciendo énfasis en el transporte terrestre de pasajeros.

### **2.2 CONCEPTOS Y BASES TEÓRICAS**

Hacia los años de 1998, Haupt propone una clasificación de algoritmos de optimización basada en las características que presentan diferentes problemas. En concreto, propone 6 criterios para clasificar los diferentes problemas y las técnicas para abordarlos. 1) Naturaleza de la función objetivo, 2) Número de variables, 3) Dependencia del tiempo, 4) Naturaleza de las variables, 5) Existencia de restricciones y características de las mismas, y 6) Métodos de búsqueda. Estos se detallan en *García*<sup>2</sup>.

En el 2004, Michalewicz y Fogel, proponen una clasificación en dos grandes grupos. Por un lado, identifican las técnicas tradicionales y, por otro, las técnicas heurísticas. Los métodos tradicionales incluyen la búsqueda exhaustiva, la búsqueda local, la programación lineal, la programación no lineal, algoritmos de tipo ávido, la programación dinámica y los métodos de ramificación y acotación.

---

<sup>2</sup> García A. Programación del transporte de hidrocarburos por oleoductos mediante la combinación de técnicas metaheurísticas y simulación. Universidad politécnica de Madrid. 2007

Los heurísticos constituyen una segunda forma de abordar problemas de optimización. Son sencillos de aplicar, fáciles de entender y, si se diseñan de forma adecuada, ofrecen soluciones aceptables en tiempos mucho menores que los métodos exactos.

Por último, un tercer y gran grupo lo forman las técnicas metaheurísticas, que cuentan con varias aplicaciones para un conjunto muy amplio de áreas. En este capítulo nos centraremos en el conocimiento a fondo de las metaheurísticas más usadas, y como se han utilizado en la solución de problemas multiobjetivo.

### **2.3 TÉCNICAS METAHEURÍSTICAS**

Un problema de optimización se resume en hallar un valor máximo o mínimo de una función dada, la cual es la representación general del problema. Las teorías clásicas de optimización como el cálculo diferencial, tratan el caso en que las posibles instancias de entrada de éste son infinitas.

Los problemas de optimización combinatoria generalmente tienen un dominio finito de posibles soluciones, y puede parecer que resolver este tipo de problemas es más sencillo que el caso anterior, ya que se puede buscar solución por solución la más conveniente. En otras palabras, se podrían generar todas las soluciones posibles, evaluarlas según el costo asociado que tengan y obtener la mejor solución. Sin embargo, aunque un método así en teoría siempre llega a la solución óptima, es demasiado costoso desde el punto de vista de la eficiencia, pues el tiempo que se utilizará en calcular y evaluar las posibles soluciones crece de manera exponencial al aumentar las entradas del problema.

Existen problemas combinatorios para los que no se conocen algoritmos que los resuelvan, además provocan que el tiempo de cálculo crezca exponencialmente cuando crece el tamaño del problema.

Por otro lado existen problemas que cuentan con algoritmos que los resuelven de forma eficiente, ya que el tiempo de cómputo crece de forma polinomial cuando aumenta el tamaño del problema. Como fue expuesto por *Barril*<sup>3</sup> se ha probado que la mayoría de problemas de optimización pertenecen a la clase NP.

Todos los problemas que pertenecen a la clase NP pueden ser reducidos polinomialmente a todos los demás, es decir si existe una solución en tiempo polinomial para uno de ellos, se podría dar para todos los problemas de NP. Se dice que un problema es NP-duro, si cualquier problema en NP es polinomialmente transformable en él, aunque el problema en sí no pertenezca a NP. Si el problema además pertenece a NP, se lo denomina NP-completo.

En Inteligencia Artificial se emplea el calificativo de Heurístico en un sentido muy genérico, para aplicarlo a todos aquellos aspectos que tienen que ver con el empleo de conocimiento en la realización dinámica de tareas. Si bien los estudios sobre Heurísticas son cada vez más avanzados, es bueno recalcar que son procesos que se basan en el sentido común, es decir, siguen un patrón lógico y se potencian con la capacidad de procesamiento de los equipos actuales de computación. Existen algunos factores que pueden producir que utilizar Heurísticas para la resolución de un problema, sea atractivo, ver *Barril*<sup>3</sup>

Una de las ventajas de utilizar Heurísticas es que pueden generar más de una solución, así se puede flexibilizar la utilización de una u otra dependiendo de cual se adapta mejor a la realidad de quien la quiera. Esto puede ocurrir cuando no se han podido modelar restricciones o bien se ha simplificado el modelo original. Existe una clasificación de Heurísticas, dependiendo el modo en que buscan y construyen sus soluciones, es así como se separan en:

---

<sup>3</sup> Barril J. Algoritmos meméticos y su aplicación en fixtures deportivos. pontificia universidad católica de valparaiso.2005

- Métodos Constructivos: añaden paulatinamente componentes individuales a la solución, hasta que se obtiene una solución factible.
- Métodos de mejora local: parten de una solución factible y mediante operaciones sobre la solución, se llega a mejorarla hasta que se cumpla un determinado criterio de parada.

El término Metaheurística quiere decir “más allá de la Heurística”, y formalmente es un proceso maestro iterativo, que guía y modifica operaciones Heurísticas subordinadas para producir, de forma eficiente, soluciones de alta calidad. Las metaheurísticas se emplean para abordar problemas de gran complejidad y, típicamente, de carácter combinatorio. A diferencia de los métodos únicamente heurísticos los cuales son sencillos e intuitivos pero, a cambio, ofrecen soluciones en general pobres, las metaheurísticas permiten obtener soluciones suficientemente buenas en tiempos razonables mediante estrategias de búsqueda que tratan de huir de óptimos locales.

### **2.2.1 CLASIFICACIÓN DE METAHEURÍSTICAS**

Como las metaheurísticas son estrategias para diseñar heurísticas, se pueden dividir en tipos según la forma en que se quiere lograr una solución de buena calidad. Algunos de los tipos fundamentales de Metaheurísticas son los de 1) Relajación, 2) Constructivas, 3) de Búsqueda Local y 4) Evolutivas o iterativas. Ver *Barri<sup>4</sup>*

Podemos encontrar Metaheurísticas que contienen características de varios de los tipos antes descritos. Entre ellas se encuentran las metaheurísticas de descomposición y las de memoria a largo plazo. Las primeras descomponen el problema en subproblemas y los resuelven por separado. Las de memoria a largo

---

<sup>4</sup> Barril J. Algoritmos meméticos y su aplicación en fixtures deportivos. pontificia universidad católica de valparaiso.2005

plazo son heurísticas de aprendizaje, y utilizan la información de las soluciones de alta calidad que se generan para ajustar los criterios de selección futura.

## **2.2.2 TECNICAS METAHEURISTICAS PARA OPTIMIZACION DE PROBLEMAS COMBINATORIOS**

A continuación se presentaran algunas metaheurísticas, las cuales han sido las más empleadas y son muy interesantes para abordar la solución del problema de transporte terrestre que presentamos en este trabajo.

### **2.2.2.1 ALGORITMOS GENÉTICOS O EVOLUTIVOS**

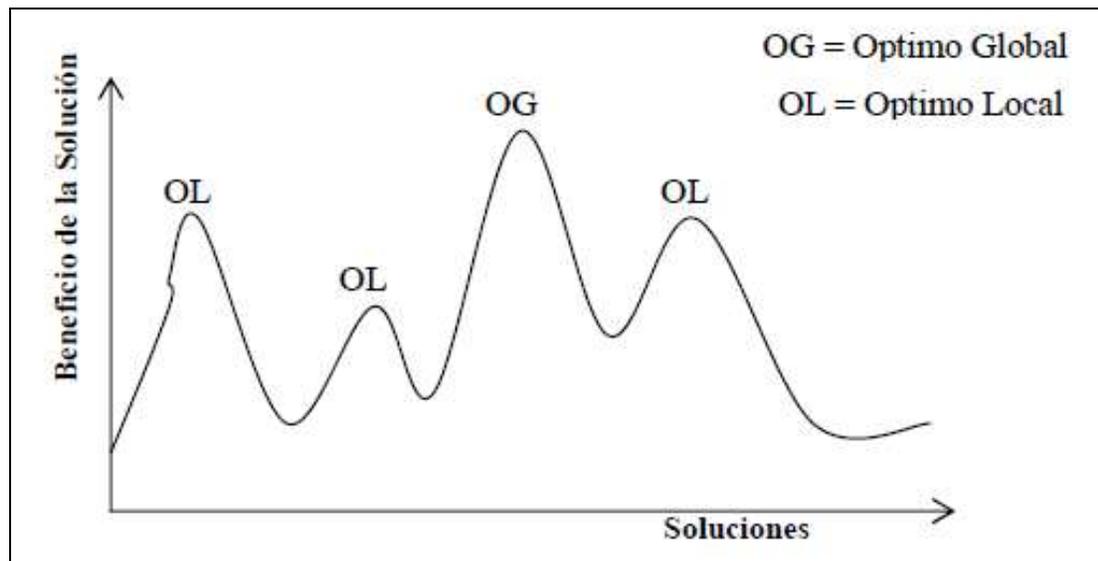
Los métodos de optimización y busca estocástica basados en modelos de evolución biológica natural, han tenido un creciente interés en las últimas décadas debido a la posibilidad que ofrecen en la resolución de problemas complejos, áreas de optimización y aprendizaje de máquina.

Los Algoritmos Evolutivos (AE), se rigen por principios del mundo biológico y se basan en la teoría de evolución de Darwin. Los AE intentan imitar algunos de los mecanismos evolutivos para resolver problemas que requieren adaptación, búsqueda y optimización.

En los AE, los puntos en el espacio de búsqueda de soluciones se modelan a través de individuos que integran un ambiente artificial. Los individuos de esta población son tomados en cada iteración, con el fin de que se relacionen y compitan, y así lograr una evolución en la población. La selección de un individuo de la población depende de una función de aptitud, que consiste generalmente en una función objetivo que se desea maximizar o minimizar para solucionar un problema de optimización determinado.

Los AE son útiles para búsquedas globales, donde los métodos determinísticos pueden llevar a máximos o mínimos locales. Por esto los AE son aptos para resolver un amplio espectro de problemas no lineales, discontinuos, discretos, multivariados, entre otros. La figura a continuación ejemplifica de manera gráfica los conceptos de óptimos locales y óptimo global.

**Figura 2. Óptimo global y óptimo local**



Fuente: Barri<sup>5</sup>

La estructura básica general de un Algoritmo Evolutivo es la siguiente:

1. Inicialización aleatoria de la población.
2. Evaluación de la función objetivo.
3. Selección de individuos más aptos de acuerdo a la estrategia de selección.
4. Aplicación de operadores de Recombinación y mutación.
5. Generación de una nueva población de soluciones candidatas.
6. Repetición de pasos 2 al 5 hasta satisfacer una condición de parada.

---

<sup>5</sup> Barril J. 2005. algoritmos meméticos y su aplicación en fixtures deportivos.

Más adelante en el capítulo 4, se diseñara toda la estructura de un algoritmo genético híbrido, y se explicará en detalle la estructura y los métodos de selección, evaluación y evolución.

#### **2.2.2.2 BÚSQUEDAS TABÚ**

La búsqueda tabú puede verse como una metaheurística que se superpone a una técnica de búsqueda y que se encarga de evitar que dicha técnica caiga en óptimos locales prohibiendo o penalizando ciertos movimientos. El propósito de clasificar ciertos movimientos como prohibidos, o "tabú", es para evitar que se caiga en ciclos durante la búsqueda. Un movimiento pierde su status de prohibido después de un período de tiempo relativamente corto, volviéndose nuevamente accesible.

La búsqueda tabú intenta emular este mecanismo fundamental de la ingenuidad humana, pero sin utilizar elementos aleatorios, sino asumiendo que no hay razón para escoger un movimiento que nos lleve a una peor solución, a menos que estemos tratando de evitar soluciones ya examinadas. Con esta sola excepción, la técnica buscará, el mejor movimiento posible de acuerdo a la métrica utilizada. Esto hace que la técnica se dirija inicialmente de forma directa hacia un óptimo local, pero la búsqueda se reinicializará manteniendo la capacidad inicial de identificación del mejor movimiento posible.

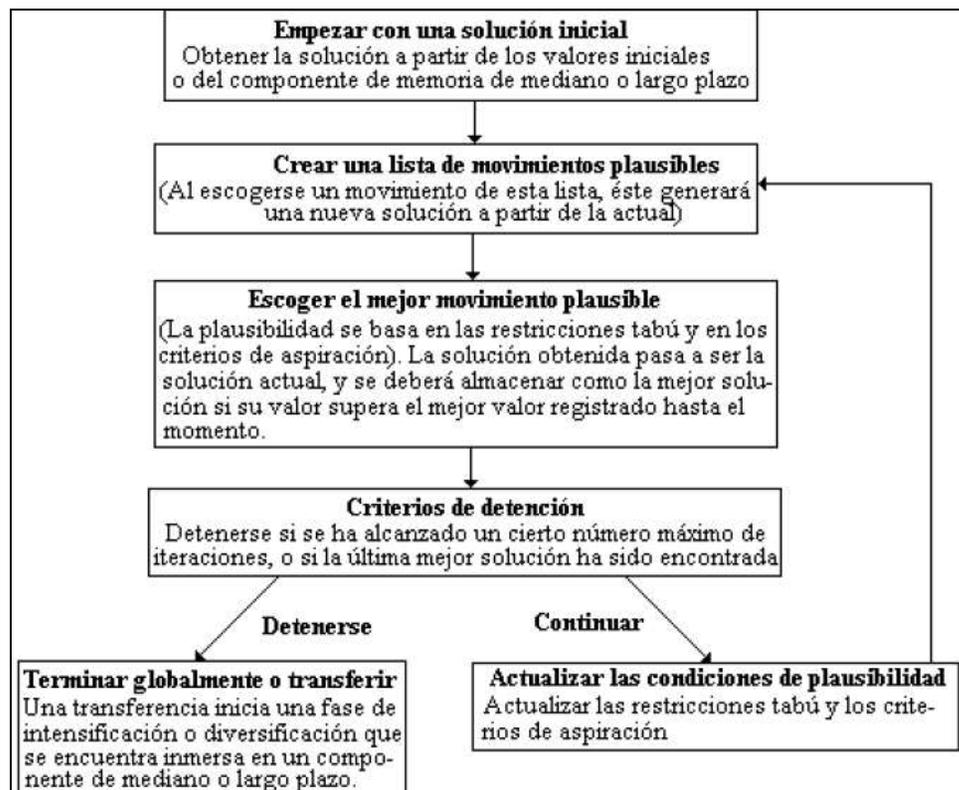
La búsqueda tabú se encuentra fundamentada en 3 puntos principales:

1. El uso de estructuras flexibles de memoria basadas en atributos, diseñadas para permitir una mejor explotación de los criterios de evaluación y la información histórica de la búsqueda.
2. Un mecanismo asociado de control basado en la interacción entre las condiciones que limitan y hacen más flexible el proceso de búsqueda. Este mecanismo se encuentra inmerso en la técnica, en forma de restricciones y

*criterios de aspiración*, los cuales permiten que un movimiento pierda su status de "tabú" debido a que proporciona una mejor solución que la actual.

3. La incorporación de memorias de diferente duración (de corto a largo plazo), para implementar estrategias que intensifiquen y diversifiquen la búsqueda. Las estrategias de intensificación refuerzan las propiedades de las combinaciones de movimientos que han demostrado, históricamente, ser buenas, mientras que las estrategias de diversificación dirigen la búsqueda hacia nuevas regiones del espacio de soluciones factibles.

**Figura 3. Proceso para memoria de Corto plazo en búsqueda Tabú.**



Fuente. Coello<sup>6</sup> Página 3

La parte medular de la búsqueda tabú es el proceso de memoria de corto plazo, y muchas de las consideraciones estratégicas en que se fundamenta este proceso

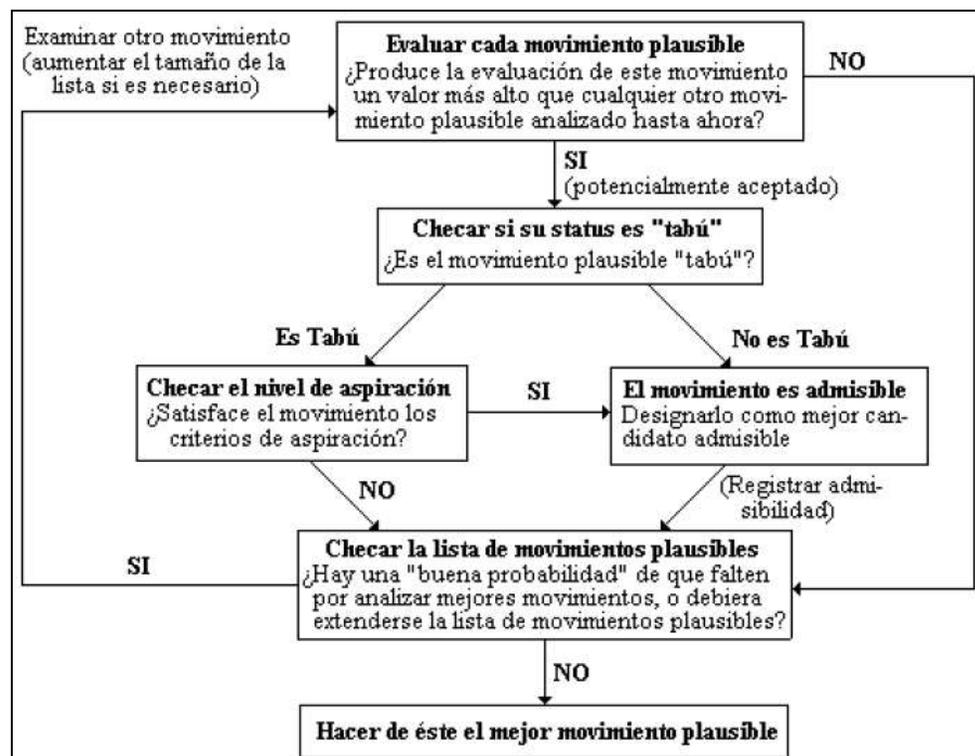
<sup>6</sup> Coello C. Búsqueda tabú: evitando lo prohibido. 2007

reaparecen amplificadas en los procesos de memoria de largo plazo. La memoria de corto plazo de la búsqueda tabú constituye una forma de exploración agresiva que intenta realizar el mejor movimiento posible sujeto a las restricciones impuestas por el problema.

Un paso crítico involucrado en la orientación agresiva de la memoria de corto plazo es la selección del mejor movimiento plausible desde un punto cualquiera. La siguiente figura ilustra esquemáticamente este mecanismo de selección.

La búsqueda tabú ha encontrado innumerables aplicaciones en muy variadas áreas dentro de la investigación de operaciones durante los últimos años.

**Figura 4. Proceso de selección de mejor movimiento posible en tabú.**



Fuente. Coello<sup>7</sup>. Página 5

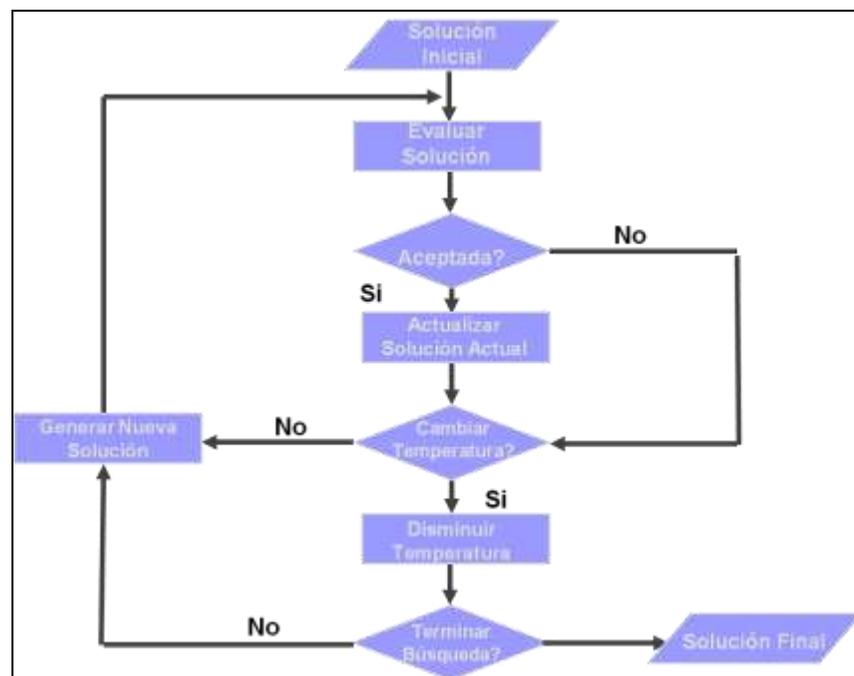
<sup>7</sup> Coello C. Búsqueda tabú: evitando lo prohibido. 2007

### 2.2.2.3 RECOCIDO SIMULADO (SIMULATED ANNEALING)

Simulated annealing (SA) o recocido simulado es un algoritmo de búsqueda metaheurística para problemas de optimización global. El nombre e inspiración viene del proceso de recocido del acero, una técnica que consiste en calentar y luego enfriar controladamente un material para aumentar el tamaño de sus cristales y reducir sus defectos. El calor causa que los átomos se salgan de sus posiciones iniciales (un mínimo local de energía) y se muevan aleatoriamente; el enfriamiento lento les da mayores probabilidades de encontrar configuraciones con menor energía que la inicial.

La técnica se divide en etapas. A cada etapa le corresponde una temperatura menor que la que tenía la etapa anterior. Por lo tanto hace falta un criterio de cambio de la temperatura (“cuánto tiempo” se espera en cada etapa para dar lugar a que el sistema alcance su “equilibrio térmico”).

**Figura 5. Estructura de operación del Simulated Annealing**



Fuente: Propia

Para ver en detalle la estructura y los parámetros neurálgicos del recocido simulado, ver Vázquez<sup>8</sup>

#### **2.2.2.4 GRASP**

La técnica GRASP (del inglés *Greedy Randomized Adaptive Search Procedure*), propuesta por Feo y Resende (1995), es una técnica metaheurística que opera en dos fases: en la primera, construye la solución y en la segunda se realiza un proceso de búsqueda local en torno a la solución obtenida en la primera.

*“La primera fase de construcción, se genera una solución obtenida tras la ejecución de diferentes etapas, en cada una de las cuales se completa parcialmente la solución hasta llegar a una solución completa. En cada etapa se dispone de un conjunto de elementos candidatos para completar la solución, estos elementos están contenidos en lo que se denomina lista restringida. Esta lista está formada por elementos que a priori pueden ofrecer buenas soluciones (aspecto ávido –greedy- del procedimiento). De entre los elementos candidatos se elige uno de forma aleatoria (aspecto aleatorio). Tras la fase constructiva se modifica la lista restringida de candidatos (aspecto adaptativo).”<sup>9</sup>*

Una vez obtenida la solución tras la primera fase, se define un vecindario y se realiza una búsqueda local, en la que solo se realizan movimientos que mejoran la solución, que finaliza cuando no existe ninguna solución mejor. Las técnicas GRASP, son técnicas intermedias entre los algoritmos totalmente ávidos o greedy, que sería equivalente a definir una lista de candidatos que contuvieran un solo elemento, y una búsqueda totalmente aleatoria, en la que la lista de candidatos incluye todos los elementos restantes para construir la solución. En la medida en que se reduce el número de movimientos candidatos, la varianza de los valores de la función objetivo de las soluciones obtenidas en la fase constructiva disminuye,

---

<sup>8</sup> Vázquez M. Recocido simulado: un algoritmo para la optimización de estructuras. Escuela técnica superior.

<sup>9</sup> García A. Programación del transporte de hidrocarburos por oleoductos mediante la combinación de técnicas metaheurísticas y simulación. universidad politécnica de Madrid. 2007

pero es más probable que la calidad de las soluciones obtenidas en la fase de optimización local disminuya. Si, al contrario, el procedimiento se hace cercano a la búsqueda aleatoria, las varianzas aumentan mucho y hay menos garantías de comenzar la fase de optimización con soluciones suficientemente atractivas.

Las técnicas GRASP ofrecen buenos resultados para ciertos problemas y presenta la ventaja de tener pocos parámetros, lo cual hace que su diseño y aplicación sean sencillos.

### **2.2.2.5 COLONIA DE HORMIGAS**

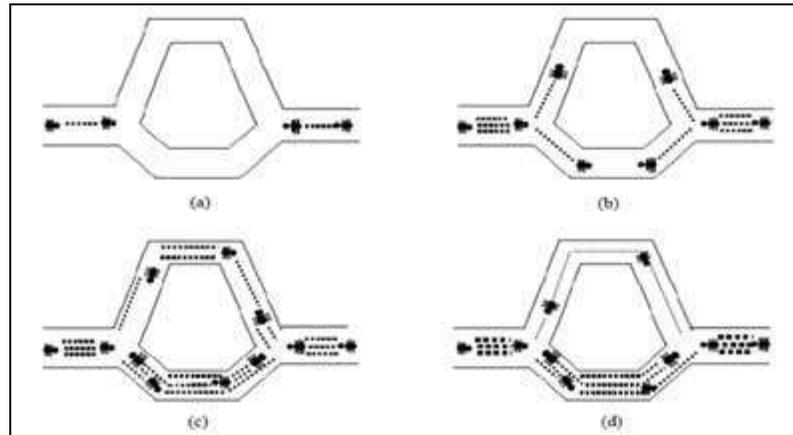
Los algoritmos de optimización de Colonias de Hormigas (OCH) se inspiran directamente en el comportamiento de las colonias reales de hormigas para solucionar problemas de optimización combinatoria. Se basan en la naturaleza de las colonias de hormigas artificiales (unos agentes computacionales simples que trabajan de manera cooperativa y se comunican mediante rastros de feromona artificiales). Los algoritmos de OCH son esencialmente algoritmos constructivos. Cada hormiga construye una solución al problema recorriendo puntos de construcción, en la cual, se emplea una lógica parecida a la de las hormigas al desplazarse empleando el rastro de feromonas.

*“Cada paso o ruta que pueda seguir una hormiga, representa los posibles pasos que la hormiga puede dar, tiene asociada dos tipos de información que guían el movimiento de la hormiga:*

- *Información heurística, que mide la preferencia heurística de moverse desde el punto  $r$  hasta el nodo  $s$ , o sea, de recorrer la arista  $a_{rs}$ . Se nota por  $\eta_{rs}$ . Las hormigas no modifican esta información durante la ejecución del algoritmo.*
- *Información de los rastros de feromona artificiales, que mide la “deseabilidad aprendida” del movimiento de  $r$  a  $s$ . Imita a la feromona real que depositan las hormigas naturales. Esta información se modifica durante la ejecución del*

algoritmo dependiendo de las soluciones encontradas por las hormigas. Se nota por  $\tau_{rs}$ .<sup>10</sup>

**Figura 6. Comportamiento de colonias de hormigas naturales**



Fuente: Alonso<sup>10</sup>

Al final, tras conocer la calidad de las soluciones obtenidas por cada una de las hormigas, se modifican el rastro de feromonas asociado a los cambios de estado de acuerdo con algún criterio que premie las transiciones correspondientes a hormigas que han ofrecido buenos resultados. La probabilidad de cada movimiento se construye como una ponderación de los dos factores explicados en la referencia anterior.

La técnica del rastro de feromonas, constituye la memoria del proceso de búsqueda, premiando que las hormigas realicen transiciones similares a las que han ofrecido buenos resultados en el pasado. La elección de cada movimiento no es determinística y esto es lo que permite, si se ajustan adecuadamente los parámetros, abrir la posibilidad de explorar nuevas soluciones en todas las iteraciones. Existen muchas variantes según la manera de construir la solución, la manera de actualizar el rastro de feromonas, etc.

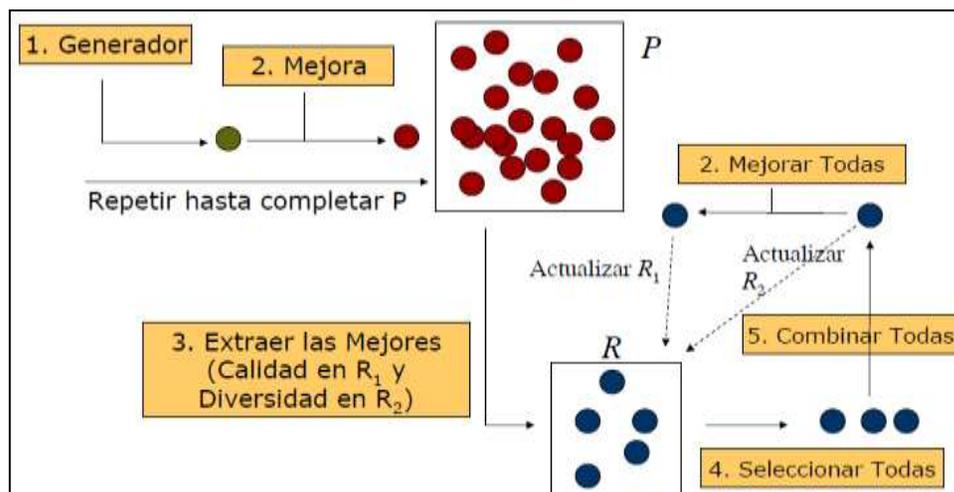
<sup>10</sup> Alonso S, Cerdón O, Fernández de Viana I, Francisco Herrera. La metaheurística de optimización basada en colonias de hormigas: modelos y nuevos enfoques. ingeniería informática. Granada, España.

### 2.2.2.6 BÚSQUEDA DISPERSA

La búsqueda dispersa fue propuesta por Glover (1977) como una técnica basada en la combinación de reglas para la programación de la producción y en la combinación de restricciones en problemas de programación entera y no lineal. La búsqueda dispersa combina soluciones para obtener otras mejores y mantiene un conjunto de soluciones de referencia que son las que utiliza para obtener las nuevas soluciones.

El método se basa en la utilización de una pequeña población, conocida como conjunto de referencia, cuyos elementos se combinan de forma sistemática para la creación de nuevas soluciones. Además, estas nuevas soluciones pueden pasar por una fase de mejora consistente en la aplicación de una búsqueda local. El conjunto de referencia es inicializado a partir de una población inicial,  $P$ , compuesta por soluciones aleatorias lo más dispersas posibles, y es actualizado con las soluciones resultantes de la fase de mejora. En esta fase, para obtener nuevas soluciones se combinan las soluciones de referencia  $R'$ , que son un subconjunto de  $P$ . Las soluciones se modifican o mejoran para garantizar que son soluciones factibles del problema.

**Figura 7. Estructura del proceso de la búsqueda dispersa**



Fuente: Propia

Este procedimiento finaliza con la actualización de R, seleccionando un conjunto de las mejores soluciones obtenidas por combinación y otras por la mejora. La técnica está construida de tal forma que el conjunto de soluciones de referencia R contiene información útil sobre las características de las mejores soluciones.

### **2.2.3 TECNICAS METAHEURÍSTICAS APLICADAS A PROBLEMAS DE OPTIMIZACIÓN MULTIOBJETIVO**

A continuación se detallan conceptos de optimización multiobjetivo y cómo varían las técnicas propuestas al aplicarlas a la aplicar estos conceptos.

#### **2.2.3.1 OPTIMIZACIÓN MULTIOBJETIVO**

Gran parte de los problemas del mundo real implican la optimización simultánea de varios objetivos que generalmente presentan conflictos entre ellos; es decir, la mejora en uno conduce a un deterioro en el otro. La presencia de tales tipos de problemas es tan significativa, que consume gran parte de nuestro tiempo cotidiano de decisión. Se trata, por ejemplo, de escoger el medio ideal para llegar al trabajo, establecer el orden de nuestras tareas, elegir el restaurante para el almuerzo, hacer las compras en el supermercado, preparar la cena y la distribución de actividades en el tiempo de ocio restante. También es el mismo tipo de problemas que enfrentan los ingenieros y técnicos a la hora de diseñar e implementar sistemas de todo tipo: existen múltiples objetivos a cumplir y se espera lograrlos todos en la medida de lo posible.

En los últimos años, la optimización de problemas que requieren optimizar más de una función, han recibido una gran atención, estando este interés motivado en gran medida por la naturaleza multiobjetivo de los problemas del mundo real.

Un problema de optimización multiobjetivo (*MOPs*, por sus siglas en inglés: *Multiobjective Optimization Problem*), general incluye un conjunto de  $n$  parámetros

(variables de decisión), un conjunto de  $k$  funciones objetivo, y un conjunto de  $m$  restricciones. Las funciones objetivo y las restricciones son funciones de las variables de decisión. Luego, el MOP puede expresarse como:

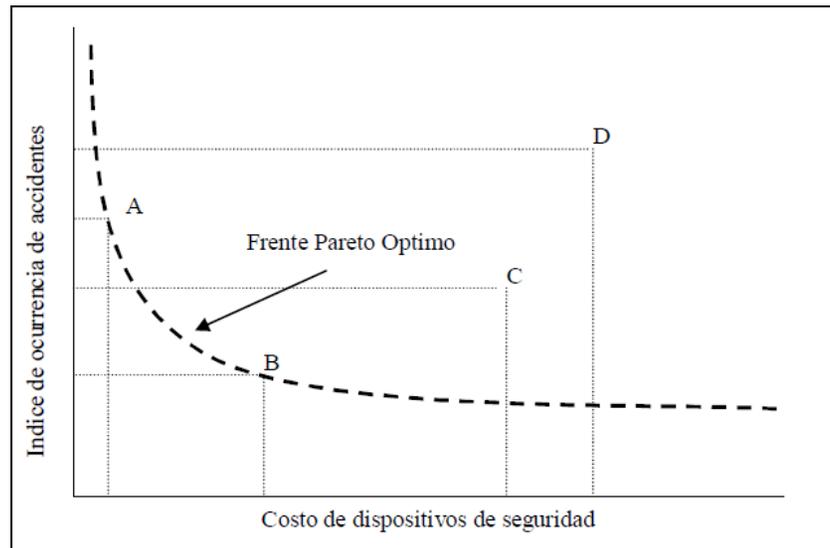
$$\begin{aligned}
 & \text{Optimizar } y = f(x) = (f_1(x), f_2(x), \dots, f_k(x)) \\
 & \text{Sujeto a } e(x) = (e_1(x), e_2(x), \dots, e_m(x)) \geq 0 \\
 & \text{Donde } x = (x_1, x_2, \dots, x_n) \in X \wedge y = (y_1, y_2, \dots, y_k) \in Y
 \end{aligned}
 \tag{2.1}$$

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

De forma general, la optimización multiobjetivo no se restringe a encontrar una única solución dado un problema, sino en proporcionar un conjunto de soluciones conocido como *soluciones no-dominadas*. Cada solución en este conjunto se dice que es un óptimo de Pareto, y cuando se representan en el espacio de objetivos, se conocen colectivamente como *Frente de Pareto*. Obtener el frente de Pareto es el principal objetivo en la optimización multiobjetivo. Esto significa que los algoritmos multiobjetivo necesitan explorar grandes porciones del espacio de búsqueda debido a que no se busca una única solución, sino un conjunto de óptimos de Pareto.

Para ilustrar este concepto se presenta la figura 8, que muestra la relación entre dos funciones  $f_1$  y  $f_2$ . La función  $f_1$  representa el costo de los dispositivos de seguridad de un sistema, mientras que  $f_2$  representa el índice de ocurrencia de accidentes.

**Figura 8. Ejemplo de la optimalidad Pareto en el espacio objetivo**



Fuente: Villagra<sup>11</sup>

La solución representada por el punto B es mejor que la del punto C, debido a que provee de una mayor seguridad a un costo menor. La solución B sería también la escogida en el caso de estar realizando la optimización de un solo objetivo. Más aún, todas las soluciones que se encuentran en el rectángulo delimitado por el origen de coordenadas y la solución C y por encima de la curva, son mejores a C. En la comparación entre C y A, se obtiene que, disminuyendo mucho los costos, el nivel de seguridad provisto por A es solo ligeramente peor, aunque C y A son no comparables entre ellos porque no se podría argumentar que uno es mejor que otro al considerar todos los objetivos. Si comparamos a A con B tampoco podríamos establecer que alguna de las dos sea mejor, si se considera que ambos objetivos son igualmente importantes y no se introduce alguna consideración de índole subjetiva. Sin embargo, B es claramente superior a C en ambos objetivos.

En los problemas de optimización de un solo objetivo (*SOPs*, del inglés *Single Objective Problem*) el resultado óptimo deseado está claramente definido.

<sup>11</sup> Villagra M, **Optimización Multiobjetivo con Técnicas de Inteligencia Artificial**. Página 3.

Partiendo del ejemplo anterior el objetivo sería minimizar el costo del dispositivo de seguridad del automóvil, y el resultado sería un dispositivo con menor precio. Sin embargo, esta condición no se cumple para los MOP's donde, en vez de un único óptimo, contamos con todo un conjunto de soluciones de compromiso.

A esto podemos añadir que muchos MOP's del mundo real requieren la utilización de métodos computacionalmente complejos para evaluar las funciones objetivo y sus restricciones. En este contexto, las técnicas deterministas no son aplicables, siendo necesario, por tanto, usar métodos estocásticos, entre ellos las metaheurísticas.

### **2.2.3.2 METAHEURÍSTICAS EN OPTIMIZACIÓN MULTIOBJETIVO**

En conclusión, una técnica de optimización multiobjetivo está diseñada para encontrar un conjunto de soluciones factibles que son igualmente buenas entre sí, tomando en cuenta todos los objetivos estipulados sin darle prevalencia a ningún objetivo en específico. Por otro lado, en optimización global se busca la solución óptima al problema. Se dice que las soluciones de un problema con objetivos múltiples son óptimas porque ninguna otra solución, en todo el espacio de búsqueda, es superior a ellas cuando se tienen en cuenta todos los objetivos al mismo tiempo. Al conjunto de estas soluciones óptimas se conoce como soluciones Pareto óptimas.

Al utilizar técnicas metaheurísticas para solución de MOP's, estas deben ser adaptadas para evaluar el proceso que hace clave la elección de el conjunto de soluciones factibles, como lo es el concepto de dominancia de Pareto. Para cada técnica, que fue detallada en el apartado 2.2.2 de este documento, se ha encontrado en la literatura diferentes formas en como exponen la evaluación de la dominancia y la optimalidad de Pareto. Para conocer cuál de los algoritmos tiene mejor desempeño que el otro, se han creado métricas que permiten saberlo. Los tres valores fundamentales que miden las métricas son:

- Minimizar la distancia del frente de Pareto producido por nuestro algoritmo con respecto al frente verdadero (suponiendo que lo conocemos).
- Maximizar la distribución de soluciones obtenidas, de manera que podamos tener una distribución de vectores tan uniforme como sea posible.
- Maximizar la cantidad de elementos del conjunto de óptimos de Pareto generados.

Es interesante hacer notar que el problema de las métricas es también multiobjetivo. Por ello, lo más recomendable es usar diferentes métricas para evaluar los distintos aspectos de desempeño de un algoritmo. La mayor parte de las métricas actuales presuponen que  $PF_{true}$  (Frente de Pareto Verdadero) se conoce (o se puede determinar en un tiempo razonable usando un proceso enumerativo). Si ese es el caso, podemos probar el desempeño de un algoritmo multiobjetivo comparando los frentes de Pareto producidos por nuestro algoritmo con respecto al frente verdadero y determinar a partir de eso ciertas medidas de error que indiquen la efectividad del algoritmo analizado.

Retomando el tema de la dominancia de Pareto y, como gracias a este concepto, se han derivado diferentes formas de cómo adaptar las técnicas metaheurísticas, es importante nombrar algunas implementaciones de las técnicas estudiadas anteriormente en MOP's .

#### **2.2.3.2.1 MOEA - ALGORITMOS EVOLUTIVOS MULTI OBJETIVO**

Las técnicas evolutivas son las que mayor literatura tienen a la hora de hablar de optimización multiobjetivo. Existen dos categorías dentro de los denominados MOEA (de sus siglas en inglés MultiObjective Evolutionary Algorithm). Los que aplican elitismo y los que no lo hacen. El elitismo es un proceso mediante el cual se seleccionan aquellas soluciones que son no-dominadas por el resto de las soluciones factibles.

A continuación se nombran MOEA's que no aplican elitismo:

- VEGA – Vector Evaluated Genetic Algorithm
- WBGA – Weight-Based Genetic Algorithm
- MOGA – Multiple Objective Genetic Algorithm
- NSGA – Non-Dominated Sorting Genetic Algorithm

A continuación se nombran MOEA's que si aplican elitismo:

- Rudolph's Elitist Multi-Objective Evolutionary Algorithm
- NSGA-II – Elitist Non-Dominated Sorting Genetic Algorithm
- SPEA – Strength Pareto Evolutionary Algorithm
- PAES – Pareto-Archived Evolution Strategy
- MOMGA – Multi-Objective Messy Genetic Algorithm

Las técnicas que aplican elitismo, deben ser usadas siempre que no haya métodos que garanticen la sobrevivencia de los mejores cromosomas. Dentro de estas técnicas una de las más utilizadas es el SPEA. Este tiene una serie de características, las cuales se utilizan en otras técnicas de forma aislada, que lo hace muy atractivo y muy eficiente en la consecución de óptimos de Pareto:

- Guarda las soluciones no dominadas que ha encontrado hasta el momento en una población externa, de esta manera implementa elitismo.
- Usa el concepto de dominancia Pareto, para asignar un valor de fitness a los individuos.
- Realiza clustering para reducir el número de soluciones no dominadas que tiene almacenadas, sin destruir las características del frente Pareto delineado hasta el momento.

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

- Combina las tres características arriba mencionadas en un único algoritmo.

- El fitness de un individuo está determinado por las soluciones que se encuentran en la población de no dominados solamente; de este modo, no es relevante que los miembros de la población general se dominen unos a otros.
- Todas las soluciones que forman parte del conjunto externo de soluciones no dominadas participan en la selección
- Se sugiere un nuevo método para inducir a la formación de nichos dentro de la población y así preservar la diversidad genética; este método está basado en el concepto de Pareto y no requiere el conocimiento o determinación de parámetros de distancia

Esencialmente, la diferencia entre el SPEA y el algoritmo evolutivo tradicional radica en la asignación de fitness y el mantenimiento de la población externa de dominantes. Los operadores genéticos son los tradicionales y los mismos criterios de parada que se aplican en otros algoritmos evolutivos, como la cantidad máxima de generaciones  $G_{max}$  y cambios en el conjunto de soluciones identificadas.

**Figura 9. Estructura algorítmica básica del SPEA**

```

Begin
  Generate initial population  $P$  randomly. Make  $P$  feasible according to constraints
  Generation  $G = 1$ 
  WHILE  $G < G_{max}$ 
    Calculate objective values for every solution in  $P$ 
    Find non-dominated solutions in  $P$ 
    Incorporate non-dominated solutions of  $P$  to  $P'$ 
    IF number of solutions in  $P'$  exceeds maximum number allowed
      Purge  $P'$  using clustering
    End IF
    Calculate fitness for every solution in  $P$  and  $P'$ 
    Apply selection to  $P + P'$ 
    Apply cross over and mutation to generate new population  $P$ 
    Make  $P$  feasible according to constraints
     $G = G + 1$ 
  End WHILE
  Print results in output file
End

```

Fuente: Propia

### 2.2.3.2.2 MOTS - BÚSQUEDA TABÚ MULTIOBJETIVO

El primer enfoque de optimización multiobjetivo utilizando búsqueda tabú, se aplicó en 1994 por Hertz<sup>12</sup> en el problema de formación de células. Se trataba de resolver una secuencia de un solo objetivo, teniendo subproblemas con múltiples restricciones. En estos subproblemas, cada objetivo fue considerado en un orden y optimizado de acuerdo a su importancia. Solo hasta el año 1997, Hansen, desarrolló un procedimiento MOTS que se acercaba realmente a la solución de problemas multiobjetivo con búsqueda tabú. Este trabaja con un conjunto de soluciones de búsqueda de soluciones Pareto óptimas en paralelo. Para encontrar el mejor candidato de cada iteración, la condición utilizada para la asignación de fitness, fue la media ponderada de los objetivos.

**Figura 10. Estructura Algorítmica básica de un MOTS**

```
Procedure basic MOTS (maximization of all n objectives):
for each solution  $x_i$  in X do set  $x_i$  to a random feasible solution and set  $TL_i = \{\}$ 
set  $ND = \emptyset$  and set count = 1 and set  $\pi^k = 1/n$  for all objectives k
repeat
  for each solution  $x_i$  in X do
    set  $\lambda = 0$ 
    for each solution j in X where  $f(x_j)$  is non-dominated by  $f(x_i)$  and  $f(x_i) \neq f(x_j)$  do
      set  $w = g(d(f(x_i), f(x_j)), \pi)$ 
      for all objective k where  $f^k(x_i) > f^k(x_j)$  do set  $\lambda^k = \lambda^k + \pi^k w$ 
    end
    if  $\lambda = 0$  then set  $\lambda$  to a randomly chosen vector from  $\Lambda$ 
    normalize( $\lambda$ )
    find the solution  $y_i$  which maximizes  $\lambda \cdot f(x_i)$  where  $y_i \in N(x_i)$  and  $A(x_i, y_i) \notin TL_i$ 
    if  $TL_i$  is full then remove oldest element from  $TL_i$ 
    add  $A(y_i, x_i)$  to  $TL_i$  as the newest element
    set  $x_i = y_i$ 
    if  $f(y_i)$  is non-dominated by all point in ND then implement the point  $f(y_i)$  into ND
    and update  $\pi$ 
    if DRIFT-criterion is reached then set one randomly selected solution from
      X equal to another randomly selected solution from X
    set count = count + 1
  end
until STOP-criterion is met
```

Fuente: Hertz<sup>12</sup>

<sup>12</sup> hertz, a., jaumard, b., ibeiro, c.c., formosinho filho, w.p., 1994. a multi-criteria tabu search approach to cell formation problems in group technology with multiple objectives. recherche operationelle/operations research

El procedimiento de búsqueda tabú multiobjetivo, MOTS, trabaja con un conjunto de soluciones actuales que a la vez están optimizados para lograr el frente de soluciones no-dominado. Los puntos de las soluciones iniciales son buscados para cubrir todo el frente y repetidamente para cada solución, se realiza una dirección de optimización para que tiendan a alejarse de los otros puntos mientras se mueven hacia la frontera no dominada.

Las soluciones toman o aplican un movimiento de acuerdo a una heurística de búsqueda tabú y cada solución guarda su propia lista tabú de movimientos. Para mayor conocimiento de los algoritmos MOTS y sus diferentes variantes, pueden dirigirse a *Pilegaard*<sup>13</sup>

#### **2.2.3.2.3 MOSA - RECOCIDO SIMULADO MULTI OBJETIVO**

Al igual de la técnica de búsqueda tabú multiobjetivo, Los algoritmos MOSA (de sus siglas en ingles MultiObjective Simulated Annealing Algorithm), no han tenido grandes estudios o intentos en una ampliación multiobjetivo debido a la naturaleza de su operación, que es la búsqueda de un punto. En la mayoría de los intentos, se desarrollaba una función de objetivo única mediante la combinación de los distintos objetivos en uno usando un enfoque de suma ponderada. El problema aquí es cómo elegir los pesos por adelantado.

Uno de los desarrollos hechos recientemente de un MOSA, fue gracias a *Smith*<sup>14</sup>. Aquí es usada una función de energía para aplicar la dominancia. Si el verdadero frente de Pareto está disponible, entonces la energía de una solución particular X se calcula como el número total de soluciones que domina X. Sin embargo, como el verdadero frente de Pareto no está disponible todo el tiempo una propuesta que se ha hecho para estimar la energía

---

<sup>13</sup> Pilegaard hansen, michael. Tabu search for multiobjective optimization: mots. institute of mathematical modelling. technical university of denmark

<sup>14</sup> Smith K, Everson R, Fieldsend J. Dominance measures for multi-objective simulated annealing,. in roceedings of the 2004 ieee congress on evolutionary computation

basada en la estimación actual del frente de Pareto,  $F'$ , que es el conjunto de soluciones no-dominadas mutuamente que fueron encontradas en un momento del proceso. Por lo tanto, la energía de la solución actual  $X$ , está dada por el número de soluciones del frente estimado que domina  $X$ .

**Figura 11. Estructura algorítmica de un MOSA**

```

Needs:
Initial and final temperatures:  $T_{\text{initial}}, T_{\text{final}}$ 
Cooling factor :  $C$ 

Produces:
A set of nondominated models: PARETO

Initialize the population of models:  $X = \{x_0\}$ 
Initialize the set of elites : PARETO = X
 $T \leftarrow T_{\text{initial}}$ 
while  $T \geq T_{\text{final}}$ 
  //  $X'$  is the intermediate population
   $X' \leftarrow \emptyset$ 
  for each  $x \in X$ 
     $x_{\text{mutated}} \leftarrow \text{mutation}(x)$ 
    if  $x_{\text{mutated}} < x$  then
       $X' = X' \cup \{x_{\text{mutated}}\}$ 
    else if  $x < x_{\text{mutated}}$  then
      if  $\text{rnd}() < \exp(-\text{distance}(x, x_{\text{mutated}})/T)$  then
         $X' = X' \cup \{x_{\text{mutated}}\}$ 
      else  $X' = X' \cup \{x\}$ 
    else
       $X' \leftarrow X' \cup \{x, x_{\text{mutated}}\}$ 
    end if
  end for
  PARETO  $\leftarrow$  nondominated models of the joint set  $\text{PARETO} \cup X'$ 
   $X \leftarrow \text{selection}(X')$ 
   $T \leftarrow T \cdot C$ 
end while

```

Fuente: Sanghamitra Bandyopadhyay, Sriparna Saha, Ujjwal Maulik and Kalyanmoy Deb<sup>15</sup>

Los autores han modificado la variable de decisión con un número aleatorio generado de la distribución laplaciano. Esta variable, ayuda a que se apliquen a las soluciones, cualquiera de los dos grupos distintos de ampliación factores:

<sup>15</sup> Sanghamitra B, Sriparna saha, Ujjwal maulik and kalyanmoy deb. A simulated annealing based multi-objective optimization algorithm: amosa. department of computer science and engineering, jadavpur university. India

- La ampliación de recorrido, la cual genera movimientos a una de las soluciones no dominadas propuesta dentro de un frente.
- La ampliación de localización, la cual localiza un frente más cercano al original y se guarda.

Estos factores de escala se actualizan con las iteraciones. Para conocer mejor, la aplicación de estos algoritmos MOSA y otras derivaciones, ir a *Sanghamitra Bandyopadhyay, Sriparna Saha, Ujjwal Maulik and Kalyanmoy Deb*<sup>16</sup>

#### **2.2.3.2.4 MOGRASP - GRASP MULTIOBJETIVO**

Si hay una técnica, junto con las de recocido simulado SA, que tienen poca literatura en cuanto a resolver problemas de carácter multiobjetivo o multicriterio, esa es las de técnicas GRASP. Solo existen tres casos de uso y estudio de los MOGASP en la literatura:

- Para resolver el problema de la mochila (Knapsack problem)
- Para resolver problemas de programación de maquinas, en los que se minimiza la tardanza y el consumo de energía en maquinas simples.
- Para resolver el problema del diseño de una red de transito. Aunque de este tipo solo se encontró una referencia en *Mauttone*<sup>17</sup>

Al igual que todas las metaheurísticas, MOGRASP, lo que hace es optimizar los objetivos considerando un frente de Pareto óptimo, en lugar de tomar como referencia el valor de una función objetivo. Tanto en la fase de construcción como en la fase de búsqueda local, se desarrollan procedimientos para evaluar las soluciones no dominadas. En la figura 12, se presenta una estructura básica para la construcción de un algoritmo MOGRASP, incluyendo las dos fases de operación de la técnica.

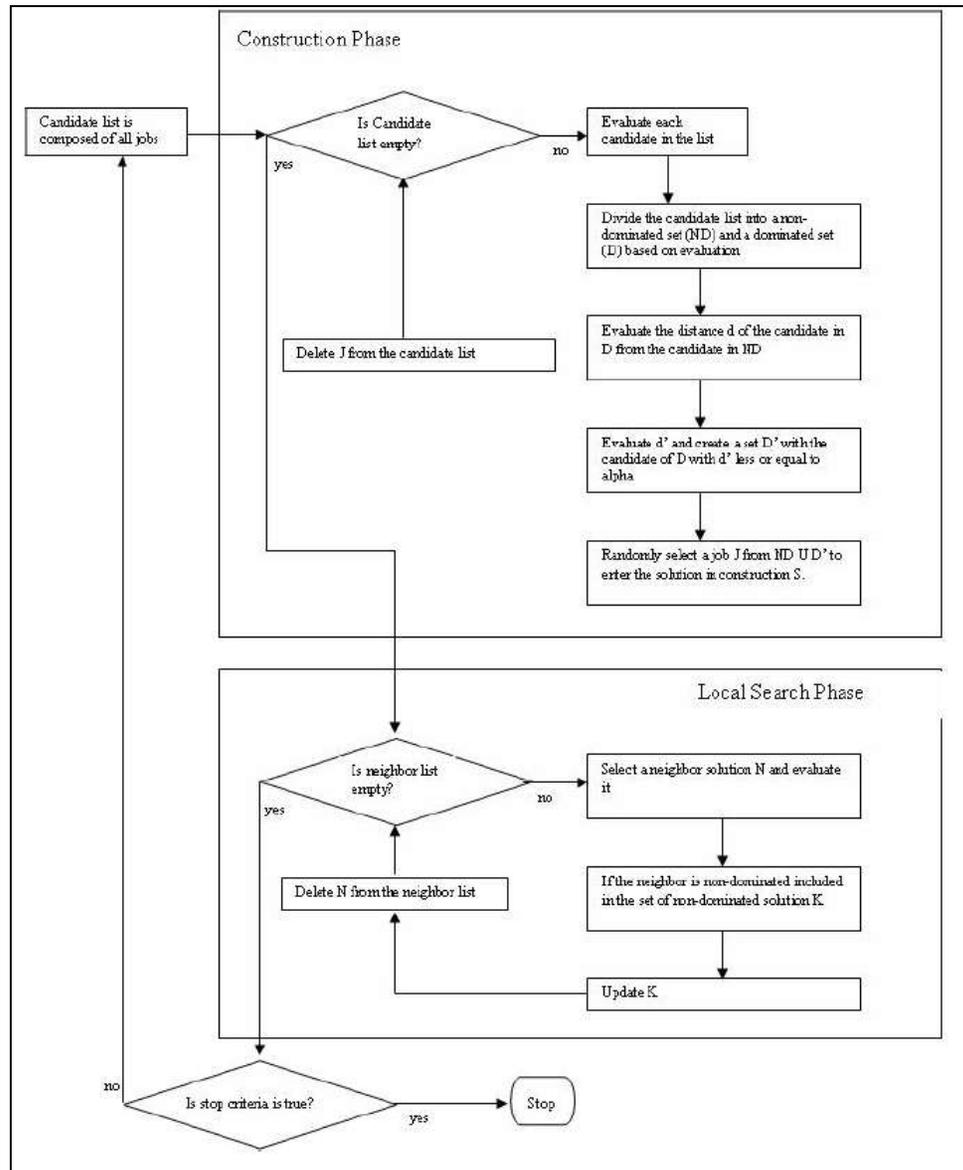
---

<sup>16</sup> Sanghamitra B, sriparna saha, ujjwal maulik and kalyanmoy deb A simulated annealing based multi-objective optimization algorithm: amosa. department of computer science and engineering, jadavpur university. india

<sup>17</sup> mauttone, antonio. optimización de recorridos y frecuencias en sistemas de transporte público urbano colectivo. universidad de la república. Uruguay. 2005

El algoritmo corre para un número predefinido de iteraciones o para un máximo tiempo de CPU que permita converger a un óptimo muy cercano a l frente de Pareto. Para conocer más sobre esta técnica ir a *Gilles Mouzon*<sup>14</sup>

**Figura 12. Estructura básica de funcionamiento de MOGRASP**



Fuente: *Gilles Mouzon*<sup>18</sup>. Página 83

<sup>18</sup> mouzon, gilles. operational methods and models for minimization of energy consumption in a manufacturing environment. Wichita state university. 2006.

### 2.2.3.2.5 MOACS - SISTEMA DE COLONIAS DE HORMIGAS MULTIOBJETIVO

Los MOACS son junto a los MOEA los de mayor estudio y mayor literatura en temas de optimización multiobjetivo, específicamente en el área de transporte, ruteo y asignación de recursos. A diferencia de todas las otras metaheurísticas, los MOACS tienen muchas derivaciones o implementaciones. La técnica básica mantiene una colonia de hormigas para cada objetivo. De esta manera en un problema con  $b$  objetivos tendremos  $b$  colonias de hormigas. Se mantiene una matriz de feromonas  $\tau$ , y cada colonia actualiza esta matriz considerando su objetivo particular asignado. La regla de transición de estados se basa en los valores de la matriz de feromonas  $\tau$ , y la visibilidad  $\eta$ . La matriz se actualiza de forma que se asigne una expresión de aprendizaje, de tal forma que la expresión denote la calidad del camino según la información aprendida.

**Figura 13. Estructura algorítmica básica de un MOACS**

```
procedure MOACO
  inicializar_parametros()
  while not condicion_parada()
    generacion=generacion + 1
    for ant=1 to m // m es la cantidad de hormigas
      construir_solucion()
      evaluar_solucion()
      actualizar_feromonas()
      actualizar_conjunto_pareto()
    end for
  end while
end

procedure construir_solucion
  sol={ }
  while existen_estados_no_visitados()
    siguiente=seleccionar_siguiete_estado()
    sol=sol U {siguiete}
    marcar_como_visitado(siguiente)
    if(actualizacion_paso_a_paso)
      actualizar_feromonas_paso_a_paso()
    end if
  end while
end
```

Fuente: Paciello , Martínez, Lezcano, Barán. <sup>19</sup>. Página 4.

<sup>19</sup> Paciello coronel, Martínez santacruz, Lezcano ríos, barán cegla. **Algoritmos de optimización multiobjetivos basados en colonias de hormigas. 2006**

Otra implementación del MOACS, está en el algoritmo BiMC (de sus siglas en inglés Bicriterion Multi-Colony), Este algoritmo extiende al MOACS original introduciendo la generalización de tener  $b$  colonias, cada una con su propia tabla de feromonas y visibilidad. Utiliza la actualización por región para realizar la actualización de feromonas, en donde la secuencia de soluciones en el frente de soluciones no dominadas de la iteración se divide en  $b$  regiones de igual tamaño. Las hormigas que encontraron soluciones en la  $i$ -ésima región actualizan la colonia  $i$ , para  $i \in [1, b]$ . Este método fuerza de manera explícita a las colonias a buscar en diferentes regiones del frente Pareto. Además de ésta, en *Paciello, Martínez, Lezcano, Barán*<sup>20</sup>, se pueden encontrar otras formas de implementación.

#### 2.2.3.2.6 MOSS – BÚSQUEDA DISPERSA MULTIOBJETIVO

La búsqueda dispersa, a pesar de haber tenido sus orígenes en los 1970s, ha tenido un gran auge en la última década, en la que, como se ha comentado con anterioridad, se ha aplicado a problemas binarios y combinatorios principalmente. En los últimos años se ha intensificado también su uso en problemas con espacios continuos y, más recientemente, en problemas de optimización multiobjetivo.

A continuación revisamos los principales trabajos reportados en la literatura que han utilizado el esquema de búsqueda dispersa para MOP's:

- SSPMO propuesto por *Molina, Laguna y Martí*<sup>21</sup>.
- SSMO propuesto por *Nebro, Luna y Alba*<sup>22</sup>.
- AbYSS es un algoritmo que utiliza conceptos del área de optimización multiobjetivo tales como la dominancia de Pareto, población secundaria, distancia de agrupamiento y un estimador de densidad. Para actualizar el conjunto de referencia se realiza una prueba de dominancia de Pareto con todos los elementos dentro del conjunto de referencia y se remueven del

---

<sup>20</sup> Paciello, Martínez, Lezcano, Barán. Algoritmos de optimización multiobjetivos basados en colonias de hormigas. 2006

<sup>21</sup> Molina, Laguna, Martí, Caballero. SSPMO: a scatter search procedure for nonlinear multiobjective optimization. volume 17, pages 111–122. *informatics journal of computing*, 2005.

<sup>22</sup> Nebro, Luna, Alba. New ideas in applying scatter search to multiobjective optimization. *third international conference, emo 2005*, volume 3410, pages 443–458, guanajuato, México, march 2005.

conjunto a aquellas soluciones que son dominadas por la nueva solución. Si la nueva solución domina a alguna otra, entonces es añadida al conjunto de referencia. La distancia de agrupamiento le permite al algoritmo calcular la densidad de dispersión de las soluciones que se encuentran en la población secundaria para cada función objetivo.

**Figura 14. Estructura algorítmica de un MOSS**

```

Datos de entrada: Población secundaria y una población terciaria
Resultado: Población secundaria (una aproximación al conjunto de óptimos de pareto)
repeat
  ObtenerSolucionesDispersas()
  for n = 1 to MaxSolucionesDispersas do
    pl = ObtenerSolucionDispersa()
    //Método para actualizar el conjunto de referencia
    ActualizarConjuntoReferencia(pl)
    for i = 1 to TamConjRef do
      for j = i + 1 to TamConjRef do
        //Método de combinación de soluciones
         $\vec{x} = BLX - \alpha(\text{pobConjRef}(i), \text{pobConjRef}(j))$ 
        //Método de mejora
         $\vec{x} = \text{Mutar}(\vec{x})$ 
        if  $\vec{x}$  no es  $\epsilon$ -dominado then
          Agregar la solución  $\vec{x}$  a la población secundaria
        end
      end
    end
  end
end
until MaxIter

```

Fuente: Ramírez<sup>23</sup>. Página 74

<sup>23</sup> Ramírez N. Una nueva propuesta para optimización multiobjetivo basada en búsqueda dispersa. centro de investigación y de estudio avanzados del instituto politécnico nacional. México DF. 2006

## 2.2.4 TRABAJOS DE REFERENCIA EN OPTIMIZACIÓN MULTI OBJETIVO APLICADO EN EL SECTOR DEL TRANSPORTE TERRESTRE DE PASAJEROS

En la literatura especializada, se encuentra una gran base de bibliografías que proponen y solucionan múltiples modelos y escenarios de problemas de ruteo. Específicamente estos se refieren a la optimización, en general, de una cadena de suministros: distribución de productos, asignación de flotas (homogéneas / heterogéneas), ventanas de tiempo, etc.

Dentro del problema de transporte de pasajeros, se encuentran soluciones planteadas para ciudades o lugares urbanos establecidos. Aquí encontramos estudios de optimización multiobjetivo para los sistemas masivos de transporte público, para transporte urbano colectivo, etc. Particularmente, hay un trabajo muy parecido al planteado por esta investigación en la ciudad de Rivera (Uruguay). Este modelo se diferencia al propuesto, ya que solo se aplico dentro del sistema vial y demanda dentro de una ciudad, y no a nivel interurbano. Por otro lado, el tipo de las soluciones son totalmente diferentes, ya que las soluciones iniciales podían ser combinadas desde cero y no estaban predefinidas como se ha propuesto en este trabajo.

En *Mauttone* <sup>24</sup>se hace referencia a este estudio. Se estudió el problema de optimización de recorridos y frecuencias en sistemas de transporte público, con un enfoque de optimización combinatoria. El problema es conocido como *TNDP* (por sus siglas en ingles, *Transit Network Design Problem*), consiste en hallar un conjunto de recorridos y valores de frecuencias, de forma de optimizar los objetivos de los usuarios (pasajeros) y los operadores (empresas de transporte), en base a información geográfica y de demanda. La resolución del TNDP tiene sentido en el contexto de planificaciones estratégicas, donde existe una autoridad

---

<sup>24</sup> Mauttone A. Optimización de recorridos y frecuencias en sistemas de transporte público urbano colectivo. universidad de la república. Uruguay. 2005

reguladora, que actúa sobre determinados componentes del sistema de transporte público, en particular los trazados de los recorridos y los valores de las frecuencias. Los objetivos de los usuarios y los operadores son contrapuestos, por lo que en contextos donde existe regulación, es responsabilidad de las autoridades determinar un nivel de compromiso adecuado.

En este trabajo se propuso un modelo de optimización combinatoria multiobjetivo para el TNDP, y se propone un algoritmo para su resolución aproximada, basado en la metaheurística GRASP. El algoritmo implementado, denominado GRASP TNDP, produce un conjunto de soluciones no dominadas que representan diferentes compromisos entre los objetivos de los usuarios y los operadores. Este trabajo se aplicó solo sobre los datos de la red vial y la demanda de una ciudad, La Rivera en Uruguay.

**Figura 15. Algoritmo MOGRASP para el problema TNDP**

```

procedure GRASP_TNDP(in  $G$ , in  $D$ , in  $D_0^{min}$ , in  $D_{01}^{min}$ , in  $\rho_{max}$ , in  $t_{max}^{ini}$ ,  $t_{max}^{fin}$ ,
                    in  $\Theta$ , in  $MaxIteraciones$ , in  $\alpha$ , out  $P$ );
01  Calcular caminos más cortos entre todo par de nodos  $(i, j)$  de  $G$ ;
02   $P \leftarrow \emptyset$ ;
03  for  $i = 1$  to  $MaxIteraciones$  do
04     $t_{max} \leftarrow$  Valor aleatorio uniforme en  $[t_{max}^{ini}, t_{max}^{fin}]$ ;
05    Construccion( $G, D, D_0^{min}, D_{01}^{min}, \rho_{max}, t_{max}, \alpha, i, R$ );
06     $F \leftarrow$  Frecuencias iniciales;
07     $S \leftarrow (R, F)$ ;
08    Evaluacion( $G, D, S, tv, te, tt, D_0, D_{01}, D_{NS}, \bar{\phi}, \phi^*, F$ );
09    BusquedaLocal( $G, D, \Theta, S, P$ );
10    Eliminar soluciones dominadas de  $P$ ;
11  end for;
12  return  $P$ ;
end GRASP_TNDP;

```

Fuente: Mauttone<sup>25</sup>. Página 80.

<sup>25</sup> Mauttone A. Optimización de recorridos y frecuencias en sistemas de transporte público urbano colectivo. universidad de la república. Uruguay. 2005

Tomando este trabajo como referencia de nuestro modelo, es importante aclarar ciertos ítems:

- Nuestro modelo está definido sobre dos objetivos principales, todos con un enfoque desde el punto de vista, de obtener el mayor beneficio de la empresa transportadora, no desde la perspectiva del usuario.
- El problema que resolvemos es de planeación estratégica, al igual que lo planteo *Mauttone*, ya que se cuentan con una serie de restricciones dadas, por entidades que velan por su cumplimiento y por la misma empresa. Esto restringe el campo de acción combinatoria del problema.
- Nuestro trabajo, propone la optimización multiobjetivo utilizando un algoritmo evolutivo híbrido, no un GRASP
- *Mauttone* solo planteo y estudio su modelo para una ciudad, no para crear dichas redes de tránsito entre ciudades. Aunque a simple vista, solo sería necesario el cambio de un par de restricciones y encontrar los valores de las demandas de pasajeros entre las diferentes ciudades, las restricciones de movilización entre ellas y las posibles redes interurbanas que se pueden utilizar.

De aquí en adelante, este trabajo planteará el modelo matemático de las funciones a optimizar y las restricciones generales que deberá cumplir. Luego, en el capítulo 4, se presentara los parámetros de selección y la forma en que se escogió la metaheurística, explicando cada parte de su estructura y funcionamiento. En el capítulo 5, no centraremos en explicar las rutinas y funciones que componen nuestro modelo metaheurístico. En el capítulo 6, veremos un caso aplicado a una empresa de transporte de pasajeros interurbanos y sus resultados. Y por ultimo en el capítulo 7, se verán las conclusiones y los trabajos futuros propuestos para investigación

### 3. FORMULACION MATEMÁTICA DEL PROBLEMA

#### 3.1 DESCRIPCIÓN

En este capítulo se definirán los índices, parámetros, variables y conjuntos asociados al problema en cuestión. Se planteará un modelo matemático de optimización multiobjetivo con sus restricciones para en el capítulo siguiente modelarlo de forma algorítmica.

#### 3.2 DEFINICIÓN DE ÍNDICES Y CONJUNTOS

- Ruta = Conjunto de rutas compuestas por las agencias del sistema.
- ROD = Conjunto de planes de ruteo.
- N = Conjunto de niveles de servicio.
- L = Conjunto de rutas o-d por nivel de servicio compuestas de por lo menos de 3 agencias, donde la agencia origen inicial es igual a la agencia Destino final
- TP = Conjunto de tipo de peajes.
- RA = Conjunto de rangos para alimentación.
- CO = Conjunto de agencias que son centros operacionales.
- AT = Conjunto de agencias que pagan tasa terminal.
- $M(o,d,j)$  = Conjunto de mejores horarios de despacho de o,d
- (o,d) = Agencias del sistema. Nodos de la red de transporte que pertenecen al conjunto **Ruta**.
- n = nivel de servicio que pertenece al conjunto **N**

- $l$  = línea que pertenece al conjunto **L**
- $t, q$  = índice que indica periodos de tiempo pertenecientes al conjunto **T**
- $p$  = tipo de peajes que pertenece al conjunto **TP**
- $rt$  = índice de rangos de tiempo para alimentación que pertenece al conjunto **RA**.
- $rod$  = plan de ruteo que pertenece al conjunto **ROD**
- $j$  = índice para el conjunto de mejores horarios de despacho  $o, d$ .

### 3.3 DEFINICIÓN DE PARAMETROS Y VARIABLES

- $RUTA(l, o, d, t)$  = Variable binaria que indica si la ruta descrita desde la agencia **o** hacia la agencia **d** que pertenecen a la línea **l** fue cubierto en un instante de tiempo **t**.
- $PAS(o, d, t)$  = Cantidad de pasajeros movilizados por el arco  $(o, d)$  del conjunto **Ruta** en un tiempo **t**.
- $pve(o, d)$  = Precio de venta del recorrido del nodo **o** al nodo **d** que pertenece al conjunto **Ruta**
- $CA(l, o, d, t)$  = Costos anticipado para cubrir el trayecto desde la agencia origen **o** hasta la agencia destino **d** que pertenecen a la línea **l** del conjunto **L** en un instante de tiempo **t**.
- $Vlpeaje(l, o, d, p)$  = Valor del tipo de peaje **p** en el recorrido desde la agencia **o** hacia la agencia **d** que pertenecen a la línea **l** de conjunto **L**.
- $Bter(n, d)$  = Variable binaria que es igual a 1 si para la agencia destino **d** usada en nivel de servicio **n** aplica cobro de tasa terminal, 0 en otro caso.
- $Vlalc(d)$  = Valor de la prueba de alcoholimetría en el destino **d** que pertenece al conjunto **Ruta**.
- $Vlalis(d)$  = Valor de los alistamientos en la agencia destino **d**.
- $Vlalm(o)$  = Valor de alimentación anticipado por salir de la agencia **o**.

- $Br(l,o,d,rt)$  = Variable binaria condicional que asigna alimentación al tramo desde agencia **o** hacia la agencia **d** que pertenecen a la línea **l**, si en el tiempo de recorrido del tramo existe un rango **rt** que pertenece al conjunto **RA**.
- $Vlhosp(d)$  = Valor del hospedaje en la agencia **d**.
- $Bhosp(l,o,d)$  = Variable binaria que indica si para el recorrido desde la agencia **o** hacia la agencia **d** que pertenecen a la línea **l** aplica hospedaje.
- $Bco(d)$  = Variable binaria que indica si la agencia **d** pertenece al conjunto **CO**.
- $rdi(n,l,o,d,q)$ : Si la ruta compuesta desde la agencia **o** hasta la agencia **d** que pertenece a la línea **l** del nivel de servicio **n** está activa en el instante **q**.
- $Tac(l)$  = Tiempo acumulado de recorrido en la línea **l**.
- $Trec(o,d)$  = Tiempo de recorrido desde la agencia **o** hasta la agencia **d**.
- $Thos$  = Parámetro que indica el tiempo máximo de recorrido continuo.
- $hir(rt)$  = Hora de inicio del rango para alimentación **rt**
- $hfr(rt)$  = Hora de fin del rango para alimentación **rt**.
- $hit(l,o,d)$  = Hora de despacho de la agencia **o** hacia la agencia **d** que pertenecen a la línea **l**.
- $hft(l,o,d)$  = Hora de llegada a la agencia **d** desde la agencia **o** que pertenecen a la línea **l**
- $Maxdesp(o,d)$  = Número máximo de despachos permitidos en el tramo desde la agencia **o** hacia la agencia **d**.
- $Hmejor(o,d,j)$  = Horario de despacho **j** desde la agencia **o** hacia la agencia **d**.
- $Bmejor(o,d,j)$  = Variable binaria que indica si para el recorrido desde la agencia **o** hasta la agencia **d** fue elegido el mejor horario **j** en importancia de acuerdo a la demanda acumulada que pertenece al conjunto **M**.
- $Porcdes$  = Parámetro que indica el Porcentaje de descanso dependiendo del tiempo de recorrido entre agencias.
- $Pdif(o,d)$  = Parámetro que muestra el límite máximo de aceptación en el desplazamiento de horas de despacho entre la agencia **o** y la **d**.

### 3.4 FUNCIONES OBJETIVO

El problema plantea la solución de 2 funciones objetivo.

$ZMin_n = \text{Costos Anticipos por Nivel de Servicio}$

$$ZMin_n = \sum_{n \in N} \left[ \sum_{l \in L} \left[ \sum_{(o,d) \in Ruta} \left[ \sum_{t \in T} CA_{n,l,o,d,t} * RUTA_{n,l,o,d,t} \right] \right] \right] \quad (3.1)$$

$ZMax_n = \text{Ingresos por Nivel de Servicio}$

$$ZMax_n = \sum_{n \in N} \left[ \sum_{l \in L} \left[ \sum_{(o,d) \in Ruta} \left[ \sum_{t \in T} (pve_{o,d} * RUTA_{n,l,o,d,t} * PAS_{o,d,t}) \right] \right] \right] \quad (3.2)$$

Estas funciones fueron obtenidas de la siguiente manera:

- **Ingresos**

Estos son obtenidos con los recaudos de las ventas por los recorridos de un origen **o** hacia un destino **d** en un periodo de tiempo **t**.

$$\text{Ingreso por Nivel}_n = \sum_{l \in L} \left[ \sum_{(o,d) \in Ruta} \left[ \sum_{t \in T} (pve_{o,d} * RUTA_{n,l,o,d,t} * PAS_{o,d,t}) \right] \right] \quad (3.3)$$

- **Costos anticipos**

Los costos anticipos, son costos que se entregan antes de partir un vehículo desde un origen **o** hacia un destino **d** y cubren 1) Peaje, 2) Alojamiento de la tripulación, 3) Alimentación de la tripulación, 4) Alistamientos y trabajos de adecuación, 5) Tasa terminal y 6) Prueba alcoholimetría

$$\text{Costos Anticipo}_{l,o,d,t} = \sum_{(o,d) \in Ruta} \left[ \sum_{t \in T} CA_{l,o,d,t} * RUTA_{n,l,o,d,t} \right] \quad (3.4)$$

$$CA_{o,d,t} = \left[ \sum_{p \in TP} Vlpeaje_{l,o,d,p} + (VltasaTer_d * Bter_{l,d}) + Vlalc_d + Vlalis_d \right. \\ \left. + \sum_{rt \in RA} (Vlalm_o * Br_{l,o,d,rt}) + (Vlhosp_d * Bhosp_{l,o,d} * Bco_d) \right]$$

(3.5)

### 3.5 RESTRICCIONES

#### Valores de costos y demanda

Todos los valores asignados a la función de costos anticipos y de ingresos deben ser mayores o iguales a 0.

$$Vlpeaje_{i,o,d,p}, VltasaTer_d, Vlalc_d, Vlalist_d, Vlalm_o, Vlhosp_i \geq 0 \quad (3.6)$$

$$PAS_{o,d,t}, dem_{o,d,t} \geq 0 \quad (3.7)$$

#### Demanda

El número de pasajeros movilizados en la ruta que va desde agencia **o** hacia la agencia **d** la cual pertenece a la línea **l** en un instante de tiempo **t**, debe ser menor o igual a la demanda acumulada **dem** establecida.

$$PAS_{o,d,t} \leq dem_{o,d,t} * RUTA_{n,l,o,d,t} \quad (3.8)$$

#### Uso exclusivo de una ruta o-d en un instante de tiempo.

Para un instante de tiempo **t**, el número total de rutas asignadas desde la agencia **o** hacia la agencia **d** debe ser 1 o 0.

$$\sum_{n \in N} [\sum_{l \in L} [\sum_{(o,d) \in Ruta} RUTA_{n,l,o,d,t}]] \leq 1, \forall t \in T \quad (3.9)$$

#### Asignación de rutas no bloqueadas por cierres de vías

Las rutas deben ser asignadas de modo que durante el periodo de tiempo que va desde **t** hasta **t+Δt(o,d)** estén disponibles.

$$RUTA_{n,l,o,d,t} \leq \frac{\sum_{n \in N} [\sum_{l \in L} [\sum_{(o,d) \in Ruta} [\sum_{q=t}^{t+\Delta t(o,d)} rdi_{n,l,o,d,q}]]]}{\Delta t(o,d)} \quad (3.10)$$

#### Uso de asignación de tasa terminal para el destino **d**

Para la agencia  $d$  de destino, se debe evaluar si pertenece a un nivel de servicio que pague esta tasa.

$$B_{ter_{n,d}} = \begin{cases} 1, & d \in AT(n) \\ 0, & \text{en otro caso} \end{cases} \quad (3.11)$$

### Uso de asignación de hospedaje para el destino $d$ si pasa el tiempo límite continuo en carretera

Si se cumple que el tiempo acumulado de recorrido es mayor o igual  $Thos$  (Tiempo máximo permitido de recorrido continuo en carretera), debe aplicar hospedaje en el tramo  $(o,d)$  que pertenece a la línea  $l$ .

$$B_{hosp_{l,o,d}} = \begin{cases} 1, & Tac_l + Trec_{o,d} \geq Thos \rightarrow Tac_l = 0 \\ 0, & Tac_l + Trec_{o,d} < Thos \rightarrow Tac_l = Tac_l + Trec_{o,d} \end{cases} \quad (3.12)$$

### Asignación de hospedaje si no es centro operativo

Si la agencia  $d$  pertenece al conjunto de agencias que son centros operativos para el nivel de servicio  $n$ , se asigna el hospedaje para el tramo  $(o,d)$ .

$$B_{co_d} = \begin{cases} 1, & d \in CO(n) \\ 0, & \text{En otro caso} \end{cases} \quad (3.13)$$

### Asignación de alimentación de acuerdo a rangos de horarios establecidos

Se asigna el valor de alimentación si la hora de inicio del rango  $rt$  es mayor o igual a la hora de inicio del tramo  $o,d$ , o la hora de finalización del rango  $rt$  es menor o igual a la hora de finalización del tramo  $o,d$ , que pertenece a la línea  $l$ .

$$B_{r_{l,o,d,rt}} = \begin{cases} 1, & (hit_{l,o,d} \leq hir_{rt} \leq hft_{l,o,d}) \vee (hit_{l,o,d} \leq hfr_{rt} \leq hft_{l,o,d}), \\ 0, & \text{En Otro Caso} \end{cases},$$

$$\forall rt \in RA \quad (3.14)$$

### Número máximo de despachos en un O-D

Se debe cumplir que para cada tramo asignado desde la agencia o hacia la agencia d, debe haber número máximo de despachos definidos  $Maxdesp(o,d)$ . Se debe evaluar en todas las líneas de todos los niveles de servicio.

$$\sum_{n \in N} [\sum_{l \in L} [\sum_{t \in T} RUTA_{n,l,o,d,t}]] \leq Maxdesp_{o,d}, \forall (o,d) \in Ruta \quad (3.15)$$

### Asignación de horarios de despacho para cada o-d

Los primero recorridos o-d de una línea l, se asigna el mejor horario j de acuerdo a demanda. El próximo despacho de esa línea se debe asignar la mejor hora de acuerdo a la demanda de (o+1,d+1). Si la diferencia entre la mejor hora de despacho y hora de llegada a d más un porcentaje de descanso correspondiente a (o,d), es menor o igual a un parámetro de desplazamiento permitido de los despachos entre la agencia o+1 y d+1, Pdif, se asigna el mejor despacho.

$$hit_{l,o,d} = Hmejor_{o,d,j} \wedge Bmejor_{o,d,j} = 1, \text{ donde } j = \{1,2, \dots, Maxdesp_{o,d}\} \quad (3.16)$$

$$hft_{l,o,d} = hit_{l1,o,d} + Trec(o,d) \quad (3.17)$$

$$hit_{l,o+1,d+1} = \begin{cases} hft_{l,o,d} + (Trec(o,d) * 1/Porcdesc), & Hmejor_{o+1,d+1j} - (Trec(o,d) * 1/Porcdesc) + hft_{l,o,d} > Pdif_{o+1,d+1} \\ Hmejor_{o+1,d+1j}, & Hmejor_{o+1,d+1j} - (Trec(o,d) * 1/Porcdesc) + hft_{l,o,d} \leq Pdif_{o+1,d+1} \end{cases}$$

$$\forall (o,d) \in Ruta, l \in L. \quad (3.18)$$

### Cruzamientos entre O-D.

Para un tiempo de recorrido  $\Delta t(o,d)$  posterior a t, no debe ser asignado nuevamente ninguna ruta desde la agencia o hacia la agencia d en ninguna línea l ni nivel de servicio n.

$$\exists n \in N, l \in L, (o,d) \in Ruta, t \in T \mid RUTA_{n,l,o,d,t} = 1 \rightarrow \sum_{n \in N} \left[ \sum_{l \in L} \left[ \sum_{q=t+1}^{t+\Delta t(o,d)} (RUTA_{n,l,o,d,q}) \right] \right] = 0,$$

$$\forall (o,d) \in Ruta \quad (3.19)$$

## **4. SELECCIÓN, DISEÑO Y DESCRIPCIÓN DE UNA METAHEURÍSTICA PARA LA GENERACIÓN Y OPTIMIZACIÓN MULTI OBJETIVO DE PLANES DE RUTEO PARA EL TRANSPORTE TERRESTRES DE PASAJEROS.**

### **4.1 DESCRIPCIÓN**

En este capítulo, se hace una comparación general de las metaheurísticas estudiadas en el capítulo 2 con un enfoque multiobjetivo, con el fin de determinar la mejor estrategia para la construcción de nuestra técnica. Luego se describirá a fondo la estructura y funcionamiento del algoritmo que se implementará en la solución de nuestro problema.

### **4.2 COMPARACIÓN Y SELECCIÓN DE TÉCNICAS PARA LA SOLUCIÓN DEL PROBLEMA PLANTEADO.**

Siempre que nos enfrentamos a un nuevo problema de optimización surge, irremediablemente, la misma pregunta: ¿Cuál es la mejor manera de resolverlo? Para encontrar la respuesta, podemos apoyarnos a nuestros conocimientos en la materia o, mejor aún, a los del conjunto de la comunidad científica, realizando un estudio bibliográfico que nos permita identificar las mejores alternativas existentes y nos ayude a escoger una de ellas. Sin embargo, cualquiera que se haya visto en esta situación se habrá dado cuenta de que esta tarea no es tan sencilla como parece. En la actualidad se publican miles de artículos al año, donde se proponen nuevos algoritmos o variaciones sobre los algoritmos existentes y se prometen mejoras sustanciales con respecto a las herramientas desarrolladas hasta ese

momento. Identificar, de entre todas estas alternativas, la que más se adecua a nuestro problema sin ningún conocimiento previo es poco más que una quimera. Probar cada una de ellas, y sus posibles variaciones, requeriría de un tiempo del que, en muchas ocasiones, no se dispone.

Como primera medida, se descarto el diseño de un algoritmo exacto desde el principio, dada la naturaleza del problema (NP-HARD). Luego se estudiaron las diferentes opciones que nos permitieran de una forma rápida y sencilla, construir el cuerpo del plan, tomando soluciones históricas para ser evaluadas y modificadas. Como fue explicado en el planteamiento del problema en el capítulo 1, una solución debe estar formada por un conjunto de niveles de servicio (opciones o niveles de operación) y los cuales, a su vez, están compuestos por líneas predefinidas y especializadas para cada nivel de servicio. Esto ayuda a la formación del cuerpo del plan de ruteo.

**Figura 16. Estructura del cuerpo del plan de ruteo con  $n$  niveles de servicio**

<b>Nivel Servicio 1</b>	Línea 1-1
	Línea 1-2
	⋮
<b>Nivel Servicio 2</b>	Línea 1-L(1)
	Línea 2-1
	Línea 2-2
⋮	⋮
	Línea 2-L(2)
	<b>Nivel Servicio n</b>
Línea n-2	
Línea n-L(n)	

Fuente: Propia

Como primera medida, se debe armar el cuerpo del plan de ruteo, para que este sea evaluado y optimizado por la metaheurística tomando los parámetros, variables y restricciones del caso.

Todas las técnicas explicadas en el capítulo 2, comparten algunas características específicas debemos tener a la mano:

- Son métodos aproximados, no arrojan los óptimos globales, sino un conjunto posibles de ellos.
- Aceptan ocasionalmente malos movimientos, es decir, al ser métodos de búsqueda las soluciones que dan en algunas iteraciones no son necesariamente óptimos de Pareto.
- Son sencillas de usar y adaptar, lo único que se necesita es una representación adecuada del espacio de soluciones, una solución inicial y un mecanismo de exploración del espacio de soluciones. A medida que se han desarrollado estudios y mejoras, las variantes son menos sencillas que las técnicas originales.
- Son muy generales y prácticamente se pueden aplicar a la solución de cualquier problema de optimización multiobjetivo de carácter combinatorio, todo depende de la eficiencia de cómo se representan las soluciones.
- En cada iteración, las nuevas soluciones dependen de la solución de partida y la trayectoria seguida hasta ese momento, de manera que el proceso de búsqueda pase varias veces por la misma solución.

Al adentrarnos en la literatura de cada una de estas técnicas y compararlas, podemos deducir que:

- A excepción de los MOGRASP y de los MOACS, todas las técnicas son de carácter iterativo, esto es, se parte desde una solución completa, o varias de ellas, a partir de la cual(es) se obtienen soluciones también completas.
- Las diferencias entre unas y otras, radica en la forma de obtener unas soluciones a partir de otras y como hacen converger a el conjunto de no dominancia
- Los MOEA y los MOSS utilizan parámetros muy sencillos de adaptar y combinar para obtener muy buenas soluciones en tiempos bastante buenos en

comparación al resto de técnicas. Debe ser una de las razones, que junto a los MOACS, son los que más literatura especializada tienen.

Ahora bien, encontramos que si se aplican procesos de hibridación a las técnicas, aplicando heurísticas u otras metaheurísticas para ayudar a la búsqueda simultánea, en muchas ocasiones, es posible obtener mejores resultados al resolver un problema de optimización combinatoria.

La hibridación de técnicas metaheurísticas ha resultado satisfactoria en diversos campos de aplicación de la optimización combinatoria y más en casos donde la metaheurística original cae en óptimos locales. Los algoritmos híbridos han mostrado su habilidad en el ámbito de proporcionar óptimos locales de muy alta calidad. Sin embargo, se encuentran algunas aplicaciones prácticas donde aún no son claras las razones de su éxito.

A partir de la revisión bibliográfica, del planteamiento y del tipo de soluciones que buscamos, se consideró:

- No utilizar técnicas MOGRASP ni MOACS, ya que las soluciones no serían construidas desde un principio, sino que están basadas en soluciones ya construidas (Asignación de líneas predefinidas).
- No utilizar una única metaheurística multiobjetivo exclusiva, ya que en la literatura hay mucha documentación sobre el tema (aunque no es abundante en el problema de transporte de pasajeros).
- La mejor forma de representar una solución puede estar dada por los algoritmos genéticos o evolutivos, dada la naturaleza de cómo se representan las soluciones a través de cromosomas.

Esto nos llevó a concluir que, una forma de abordar el problema con éxito sería emplear un MOEA o MOGA híbrido que ayude en la exploración del espacio de búsqueda, convergiendo de una forma rápida a un conjunto de soluciones Pareto

óptimas sin caer en óptimos locales. A esta técnica a partir de ahora la llamaremos AGHMO (Algoritmo Genético Híbrido Multiobjetivo)

### **4.3 AGHMO (ALGORITMOS GENÉTICOS HÍBRIDOS MULTI OBJETIVO)**

Las primeras aplicaciones de los genéticos híbridos se remontan a finales de los años ochenta. Las mayores aplicaciones se han hecho en torno a los algoritmos meméticos. Los algoritmos genéticos híbridos (AGH) es una metaheurística evolutiva del tipo poblacional, es decir, mantiene una población de individuos llamados agentes, que representan soluciones factibles del problema determinado. Es aquí donde aparece una diferencia con el Algoritmo Genético (AG), ya que un agente puede representar más de una solución al problema. Esto porque un agente representa una evolución al ser capaz de resolver el problema de manera más eficiente que otras de las cuales aprendió.

Computacionalmente, esta evolución se logra a través de la inclusión de optimizadores locales al proceso genético, logrando que los agentes (cromosomas) sean óptimos locales, es decir, tengan una buena calidad para su posterior reproducción, lo que aumentará la probabilidad de encontrar una solución cercana al óptimo global más rápidamente. Los agentes interrelacionan entre sí en un ambiente de cooperación y competición, de manera similar a lo que ocurre en la naturaleza entre individuos de una misma especie, con el fin de realizar un paso generacional y evolucionar la población.

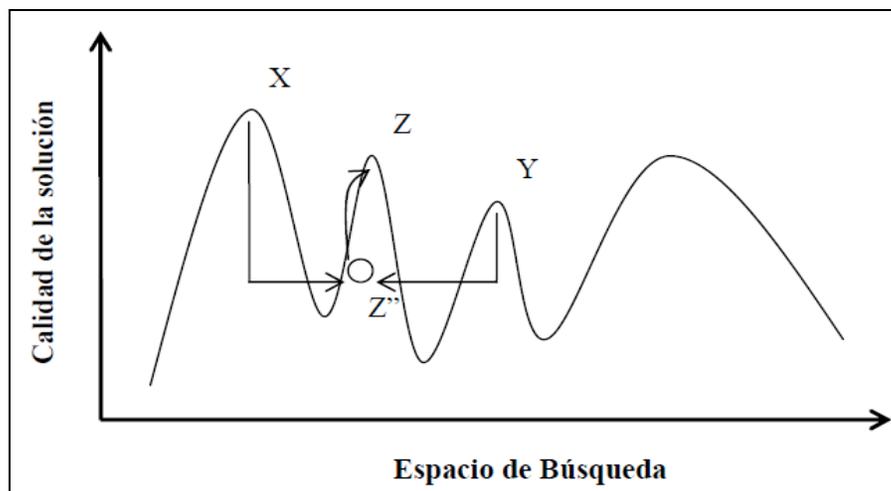
El paso generacional supone una mejora en los agentes. Esto se hace mediante operadores biológicos, recombinación y mutación, los cuales trabajan en conjunto para crear nuevos individuos (agentes), cuyas características en gran parte serán combinaciones genéticas de sus padres (dos o más agentes de la generación anterior).

### 4.3.1 ESTRUCTURA Y COMPORTAMIENTO DEL AGHMO PROPUESTO

Los algoritmos genéticos híbridos, buscan en el sub-espacio de óptimos locales dentro del espacio de búsqueda total de soluciones. Luego se toma como padres X e Y, se aplica Recombinación creando un hijo Z, el cual es reparado mediante un optimizador local con el fin de buscar un óptimo cercano a él.

Algorítmicamente las características más distintivas de un AGH son la incorporación de operadores de búsqueda local y conocimiento del problema. Los operadores de búsqueda local producen una disminución del tiempo en que se entrega una respuesta de calidad, ya que los agentes realizan una búsqueda propia en el espacio total haciendo que la población inicial contenga agentes prometedores, lo que reduce el espacio de búsqueda y acelera el proceso de encontrar una buena solución. El aprovechamiento del conocimiento del problema se logra gracias a la incorporación de operadores de recombinación inteligentes (propios del problema a resolver).

**Figura 17. Comportamiento de algoritmos genéticos híbridos (AGH)**



Fuente: *Barril Arenas*<sup>26</sup>. Página 25

<sup>26</sup> Barril J. 2005. Algoritmos meméticos y su aplicación en fixtures deportivos.

Entonces, al asignar valores a los genes de la descendencia según los valores de los padres, se está aprovechando lo calculado por la aplicación de recombinación a agentes previos, logrando óptimos locales nuevos a partir de óptimos encontrados anteriormente. Con este enfoque se tiende a subsanar la dependencia con respecto a soluciones iniciales. Si bien los descendientes toman la mayoría de los genes de alguno de sus padres, pueden tener algunos elegidos al azar.

A continuación se presenta la estructura algorítmica del AGHMO (de las siglas Algoritmo Genético Híbrido Multiobjetivo) planteado para la solución del problema de transporte terrestre de pasajeros del cual es objetivo en este trabajo.

**Figura 18. Estructura básica de AGHMO planteado**

```

AGHMO (Max-Generation, Crossover-Prob, Mutation-Prob, load_time_function)
Call _ Initial-Generation-Heuristic (&P, Options, satisfaction _ criterion);
Apply Local Optimizaltion in P Randomly to generate PI;
Num-Gen =0;
  While (Num-Gen ≤ Max-Generation)
    Make P feasible according constraints;
    Calc-function (Cost, Utility);
    Obtain-elitist-population (&New-pop); //Return a non-dominated population
      If (Num-non-dom > Max-non-dom)
        Run _ Clustering-Function (); //Reduce non-dominated population
      End If
    Calc-Strengths ();
    Calc-Fitness ();
    Roulette _ selection ();
    Crossover _ operator (&Pc, Crossover -Prob);
    Mutation _ operator (&Pm, Mutation-Prob);
    Apply Local Optimization(&PI,Pc,Pm)
    Update-Population (&P,Pc,Pm,PI);
    Num-Gen++;
  End While
End AGHMO

```

Fuente: Propia

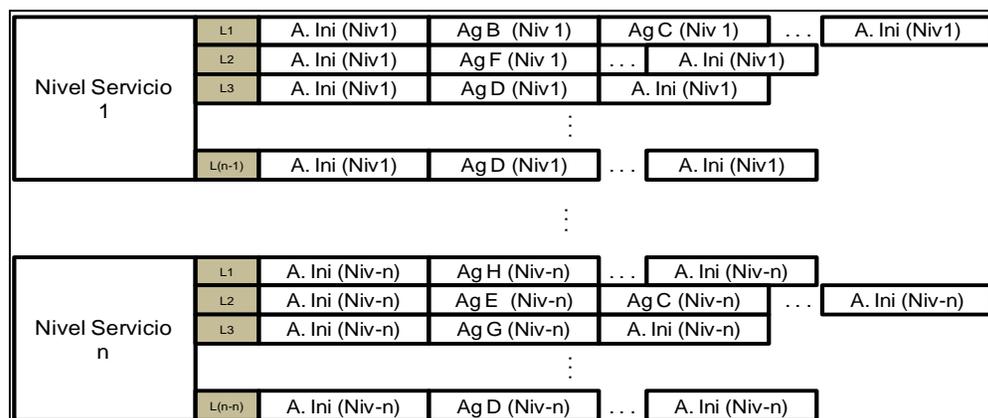
En los siguientes apartados se presentaran la descripción de la estructura funcional y el diseño del algoritmo AGHMO planteado para la solución del problema. Se describirá la estructura del cromosoma que representara una solución y las diferentes funciones y operadores que implementa la metaheurística en el proceso de optimización.

### 4.3.2 DEFINICIÓN DE CROMOSOMAS

Como lo muestra la figura 16 en el apartado 4.2, una solución generalizada se puede presentar como el conjunto de niveles de servicios operativos de la empresa transportadora, compuestos por un número de líneas L por cada nivel n de servicio que son exclusivas para él.

Cada línea está compuesta por un conjunto de agencias las cuales, de acuerdo a las restricciones, tienen libre tránsito entre ellas, es decir, puedo ir de la agencia inicial a la agencia B, luego de la agencia B puedo ir a la agencia C, así sucesivamente hasta el regreso a la agencia inicial.

**Figura 19. Estructura completa del cromosoma para una solución factible**



Fuente: Propia

Para generalizar, una solución factible (un plan de ruteo factible), debe estar compuesto por un conjunto de niveles de servicios, que a su vez, los componen líneas predefinidas por agencias o-d del sistema.

#### **4.3.3 POBLACIÓN INICIAL O SEMILLA**

Este proceso consiste en la construcción del conjunto de cromosomas que representan las soluciones del problema. Para la construcción pueden usarse Heurísticas constructiva o se pueden utilizar procesos aleatorios de generación de cromosomas. Es importante mencionar que para cada problema se pueden crear procedimientos particulares que se acomoden mejor a los requerimientos y restricciones involucrados. A continuación se describe la heurística propuesta para la construcción de la población inicial de soluciones factibles para el problema a tratar. No se escogió ninguna heurística de construcción como por ejemplo los GRASP o los ACS. Se diseño un proceso aleatorio de generación de planes factibles de acuerdo a las restricciones del problema.

##### **4.3.3.1 ESTRUCTURA GENERAL DE LA HEURÍSTICA DE GENERACIÓN DE PLANIFICACIONES SEMILLAS.**

La heurística necesita ciertas variables o parámetros de entrada para su buen funcionamiento:

- Se debe saber cuántos buses por cada nivel de servicio van a operar en el plan. Esto se utiliza para saber la cantidad de líneas programadas.
- Tiempo de duración de una línea. Este tiempo puede estar entre 1 a 5 días.
- La matriz origen – destino de cada nivel de servicio, que maneje tiempos, distancias, horas máximas y mínimas para despachar buses, el número máximo de despachos por día y rutas alternas.

Grosso modo la heurística funciona de la siguiente manera: 1) Genera un cuerpo para cada plan de ruteo, que debe estar restringido por los parámetros iniciales,

teniendo en cuenta el número de vehículos (Buses) activos y disponibles que la empresa tiene en un instante de tiempo. 2) Tomando como base este cuerpo generado, la heurística planifica cuáles líneas (ya predefinidas) son factibles de programar. 3) Se descomponen cada una de las líneas, tramo a tramo, con el fin de evaluar los tiempos, rutas principales o alternas de acuerdo a restricciones del tramo, horarios de despacho y costos que acarrea este. 4) Al final se evalúan los cruces en ruta para minimizarlos y se asigna un costo a cada tramo.

**Figura 20. Estructura de heurística para generación de planes semillas.**

```

Main (Options, satisfaction_ criterion, (Load _ Data _ Functions))
  While (Number options complete)!(/satisfaction _ criterion)
    For (Each Level N)
      Call Body_ Generation_ Function (&Num _ lines);
      For (Each Line in Level N)
        Call Planning_ Function ();
      End-For;
      Run_ Time_ Function ();
      Run_ Cross_ Function ();
      Run_ Cost _ Function ();
      End-For;
      If (Cost _ Function  $\leq$  Historical _ Cost) then
        Satisfaction_ criterion=true;
      End-If
    End While
  End Main

```

Fuente: Propia

A continuación se presentan las diferentes funciones y rutinas que componen la heurística de generación de planes semillas.

#### **4.3.3.2 FUNCIÓN PARA GENERAR EL CUERPO Y DE PLANIFICACION DE LINEAS PARA UN PLAN DE RUTEO.**

En la operación de una empresa transportadora de pasajeros, en general, se cuenta que, cada línea debe durar un numero X máximo de días. Este número varía según las condiciones dadas por cada empresa. El primer paso para

construir un plan de ruteo, es obtener los días que se estima durará cada una de las líneas. Esta duración, toma en cuenta el número de buses que la empresa posee como activos (vehículos listos para ser programados inmediatamente) y los que estarán disponibles, siendo estos con los que se cuenta por si hay imprevistos con los buses activos. Esta restricción es necesaria para no programar rodamientos que tengan más días que vehículos activos y disponibles en el momento, ya que se corre el riesgo de hacer no factible un plan.

**Figura 21. Función para generar cuerpo del plan de ruteo**

```

Void Body_ Generation_ Function (&Num_ lines)
a /* Max. # Days by Line.*/
B_ line /* Body line.*/
l /* Count number of lines*/
While (B_ line ≤ Number of days)
  Rand = Random (a) + 1;
  B_ line = B_ line + Rand;
  Option_ Level [Rand] += 1;
  l++;
End While
Num_ Lines = l;
Return Num_ Lines;
End Void

```

Fuente: Propia

Esta rutina genera de manera aleatoria el número de días que dura cada línea, tomando como restricción que no debe haber líneas de más de (**a**) días consecutivos, esto se hace para dar descanso a conductores y a los vehículos. Los parámetros pueden ser modificados de acuerdo a la empresa que lo utilice. Una vez se ha construido el cuerpo, se realiza una búsqueda aleatoria de prioridad para conseguir las líneas que se acoplen a este cuerpo. Esta base de líneas, es dada por la empresa transportadora y debe ser resultado de los años de operación de la misma (dados por la experiencia), esto con el fin de realizar planes factibles y muy deseables para el movimiento de la flota. Es importante destacar, que la función que ayuda a planificar las líneas, descarta la posibilidad de repetir líneas

de manera consecutiva, para dar paso a otras líneas que ayuden a abarcar todo el territorio sin dejar de ser factibles y óptimas para la operación. Este histórico de las líneas de operación, es administrado por la empresa, y actualizadas según las demandas en los diferentes periodos de operación.

**Figura 22. Función de planificación de líneas predefinidas para el cuerpo generado**

```
Void Planning_ Function () //Select first state "a" to assign
If (a>0)
  Restrict state "a" for future assignment.
  Select line a;
Else
  Search another state b to evaluate.
  Assign line b;
End If.
Call Decomposition_ Line_ Function (Line_ Selected); //Evaluate line by couples
  Update line's constraint
End Void
```

Fuente: Propia

Esta rutina hace uso de una función que se encarga de descomponer la línea seleccionada para poder ser evaluada por pares de agencias.

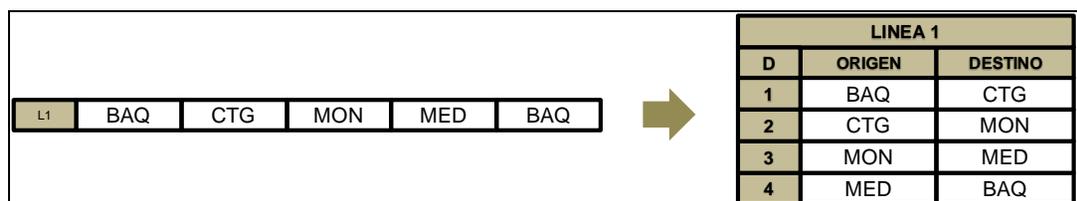
**Figura 23. Rutina de descomposición de líneas**

```
Void Decomposition_ Line_ Function (Line_ Selected line)
Int i, j, Long;
Long = line. Length ();
For (i=1; i<=Long; i++)
  If (line. Substring (i,1)==" ")
    j++;
  Else
    Assign Substring (i,1) to a new Array(j);
  End If
End For
Return New Array (j) //Decomposition Line Array
End Void
```

Fuente: Propia

Al asignar las líneas se descomponen en tramos o arreglos O-D para facilitar su evaluación y acceso. Una línea, como se verá a continuación, es un arreglo de agencias secuenciales, las cuales fueron evaluadas por un experto de la compañía, y su conformación depende mucho de la demanda que haya para los destinos que la componen. En la figura a continuación se ve como se descompone una línea:

**Figura 24. Ejemplo de descomposición de líneas**



Fuente: Propia

Después que se descompone la línea, se evalúan cada uno de sus tramos, asignándoles horarios de despacho y minimizando los posibles cruces en carretera.

#### **4.3.3.3 FUNCIÓN PARA LA GENERACIÓN DE HORARIOS DE DESPACHO Y EVALUACIÓN DE CRUZAMIENTOS**

Una vez generado el cuerpo del plan, planificarlo y descomponer las líneas, se sigue con la generación de los horarios de despacho para cada una de las rutas que operarán dentro del plan. Esta es una de las acciones más delicadas dentro del proceso de generación de planes, ya que de los horarios que se le den a cada una de las rutas, dependerá la demanda y el costo que esta genere.

La función toma el primer tramo origen – destino de la primera línea del nivel y le asigna el mejor horario de acuerdo a la demanda, si no ha sido asignado. Si ya se asigno, se busca el próximo en prioridad. Una vez se asigna esta hora, se evalúan los despachos a ser asignados para los próximos O-D de la línea.

**Figura 25. Rutina general para asignación de horarios de despacho**

```
Void Time_ Function ();  
For (Each Line in Level)  
    Call Dispatches_ Function (Line); //Evaluate Number of dispatches  
    If (First Line in Level)  
        Assign the best time in DB by O-D, if it haven't been assigned  
    Else  
        Assign next in priority;  
    End If.  
    Disp_Hor[o][d]="Assigned";  
    NMax_Desp[o][d]+=1;  
    H_sal=Disp_Hor[o][d]; //Assign first time to the line (Initial Time)  
    While (! Each O-D has been evaluated)  
        Assign times by route (O – D) depending of constraints and due to Initial  
        time.  
        Call Crossover_ Function (Hsal[O][D],O,D); //Evaluate and delete Crossover  
        between O-Ds  
    End While  
End For  
End Void
```

Fuente: Propia

Lo siguiente es sumar a esa hora de despacho el tiempo estimado de recorrido desde O hasta D para asignar una hora de llegada. Luego se aplican las restricciones, si las hay, como por ejemplo, descansos por exceso de tiempo en carretera, por horas máximas de despacho menores a la hora de llegada, entre otras (de acuerdo a la empresa). Una vez se asigna el posible despacho para el próximo O-D, se evalúa la mejor salida posible de acuerdo a demanda y que se encuentre cerca a la hora de despacho asignada por el sistema. Si esa diferencia es menor que un parámetro dado por la agencia, se asigna esa hora como la de despacho.

Con la generación de horarios, se tienen dos rutinas que ayudan a evaluar la factibilidad de una línea dentro de un plan. Estas rutinas tienen como fin modificar las líneas de un plan para que sea factible. La primera es la evaluación de número máximo de despachos por día desde un origen hacia un destino y la

evaluación de cruzamientos entre las diferentes rutas que componen las líneas de los niveles de servicio. A continuación se presentan dichas funciones.

**Figura 26. Función para evaluar número de despachos**

```
Void Dispatches_ Function (Line_ Selected line)
While (! Each O-D of the line has been evaluated)
If (Num Dispatches Acum. < Num Dispatches Accepted to the O-D)
  Num Dispatches Acum+=1;
Else
  Assign Another Target, if and only if:
  • Num Dispatches Acum (Target) < Num Dispatches Accepted to the O-D
  • Exist time between the agencies (Source – New Target)
End If
End While
  Update Line and Arrays;
End Void
```

Fuente: Propia

Esta rutina se encarga de la evaluación del número de despachos desde un origen hacia un destino. Esta función es importante porque no se debe enviar más de un número de despachos establecidos por día hacia un destino ciertos orígenes establecidos. Si la función encuentra que se ha excedido el número de despachos hacia un destino, automáticamente esta muta la línea que falló, cambiando la agencia que sobrepasó el número de despachos y asigna una en forma aleatoria de tal forma que cumpla con la condición de factibilidad, y que se pueda llegar desde el origen evaluado.

La función que evalúa los cruces, se ejecuta cada vez que se asigna un despacho. Esta analiza si ese despacho para el tramo en evaluación, presenta o no cruces con otro que se hayan asignado en un nivel de mayor prioridad. La función usa un arreglo en el que se guarda todos los despachos asignados desde O hasta D. Cada vez que se asigna un nuevo despacho se evalúa su existencia o posibilidad de cruce. Si el horario ya ha sido asignado, se modifica el horario de despacho sumando un parámetro y reevaluando el despacho.

**Figura 27. Evaluación de cruces**

```
Void Crossover_Function(Hsal[O][D],O,D)
While (sw!=1 )|(l<Length_array)
    If Cross[O][D]== Hsal[O][D]
        Sw=1;
    End if
    l++
End While
If(sw==0)
    Apply modification in Hsal[O][D] and evaluate
End if
End Void
```

Fuente: Propia

Hasta este punto, se ha descrito como se crean los planes semillas para que estos sean factibles. A partir de este punto iniciaremos la explicación del procedimiento de optimización multiobjetivo con las funciones y parámetros que implica.

#### 4.3.4 FUNCIONES Y RUTINAS DEL AGHMO

**Figura 28. Ciclo del AGHMO**

```
While (Num-Gen ≤ Max-Generation)
    Calc-function (Cost, Utility);
    Obtain-elitist-population (&New-pop); //Return a non-dominated population
    If (Num-non-dom > Max-non-dom)
        Run _ Clustering-Function (); //Reduce non-dominated population
    End If
    Calc-Strengths ();
    Calc-Fitness ();
    Roulette _ selection ();
    Crossover _ operator (&Pc, Crossover -Prob);
    Mutation _ operator (&Pm, Mutation-Prob);
    Apply Local Optimization(&Pl,Pc,Pm)
    Update-Population (&P,Pc,Pm,Pl);
    Num-Gen++;
End While
```

Fuente: Propia

La estructura del AGHMO tiene un ciclo que contiene dos criterios que debe evaluar. Lo primero es el número máximo de generaciones del algoritmo y el segundo es un criterio de satisfacción dado como parámetro. Como se puede observar, lo primero que se hace es calcular los valores de las funciones objetivo para cada una de las soluciones factibles.

#### **4.3.4.1 FUNCIÓN PARA CÁLCULO DE FUNCIONES OBJETIVO**

La rutina para el cálculo de las funciones objetivo, es simple. Lo que hace es llamar los procedimientos que evalúan los costos y la venta por planes.

**Figura 29. Función para cálculo de funciones**

```
Void Calc_ Functions (Cost, Utility)
If (Not-First Generation)
    For (Each Level)
        Call Time_ Function ();
    End If
    Run_ Cost_ Function ();
    Run_ Utility_ Function ();
End Void
```

Fuente: Propia

Como primera medida se evalúan los costos anticipo. Estos costos son todos aquellos que tienen que ver con anticipos entre las rutas y que son necesarios para la operación del vehículo. Los costos de anticipo que pueden ser evaluados son: 1) costos de alojamiento, 2) costos de alimentación, 3) costos de parqueo o uso del terminal, 4) costos de alistamiento, 5) prueba de alcoholimetría, y 6) costos de peaje. Cada agencia tiene diferentes valores de costos y diferentes reglas para ser asignado. También depende de las horas de despacho que se hayan asignado a los tramos. Estas reglas son impuestas por la empresa transportadora y los parámetros que se utilizan para la heurística, pueden ser modificados de acuerdo a las necesidades presentadas.

**Figura 30. Evaluación de costos anticipos**

```
Void Cost_ Function ()  
For (Each Line in level)  
While (! Each O-D has been evaluated)  
O=Assign a Source; D=Assign a Target;  
Load all initial cost, by O and D. // these parameters are obtained form a DB.  
Evaluate food cost, Parking cost, Accommodation cost, etc...  
Aggregate by Line;  
End While  
End For  
End Void
```

Fuente: Propia

Después de evaluar los costos, se evalúa la función de venta. La función de venta pronostica, tomando como base la información histórica de las ventas realizadas por la empresa en los días, horas y tramos del los años anteriores al año de evaluación. La función que describe y evalúa la demanda y las ventas pronosticadas es la siguiente:

**Figura 31. Función de evaluación de ventas**

```
Void utility_ function ()  
For (A Plan)  
  For (Each level and lines)  
    New_DisP {O-D} //Calculates number of dispatches by route  
  End For  
  For (All days selected)  
    While (! Each route in the line has been evaluated)  
      Select route (O-D)  
      DisP {O-D} // Number of dispatches by a O-D  
      For (All DisP {O-D})  
        Obtain sells and historic demand by each O-D  
      End For  
      Util=Historic Sells / New_DisP {O-D}  
    End While  
  End For  
End For  
End Void
```

Fuente: Propia

Los parámetros de la función de venta son las fechas entre las cuales se evaluara el plan, y las ventas y demandas que se habían presentado entre estos días en los años anteriores.

Una vez se evalúan las soluciones en las funciones objetivo, se comienza la evaluación de qué tan buena es o no una solución sobre otras. Para esto, la metaheurística hace uso de un método de jerarquización llamado fitness de la solución. Este fitness nos dará la idea de qué tan buena es una solución no dominada sobre una dominada. Pero antes debemos obtener las soluciones dominadas y no dominadas de nuestra población.

#### **4.3.4.2 FUNCIÓN DE NO-DOMINANCIA**

Es importante en toda técnica evolutiva multiobjetivo, el poder diferenciar las mejores soluciones del resto de soluciones, con el fin de purgar generación a generación aquellas soluciones que no convergen a óptimos globales. Sin embargo, estas soluciones son necesarias para aplicar los operadores de evolución descritos por un algoritmo evolutivo híbrido. 1) Selección, 2) Cruzamientos, 3) Mutaciones y 4) Meta-operador de optimización local

Como primera medida se debe obtener la población no dominada. Como su nombre lo indica, esta población es el conjunto de soluciones que no pueden ser dominadas por ningún individuo, es decir, el valor de sus funciones objetivo es el mejor dentro de todas los individuos. La búsqueda de esta población se puede hacer de dos formas, relajada o exhaustiva. La búsqueda relajada fue la implementada para la solución, y consiste en evaluar pares consecutivos y seleccionar el mejor.

Si ambas soluciones empatan se ingresa una sola de ellas a la nueva población de forma aleatoria, pero si ambas son no dominadas las dos deben ingresar a la nueva población ya que son no comparables. Esta función genera la población

elitista o la no-dominada P'. Uno de los parámetros que pueden ser modificados es el número máximo de población no dominada. Si la nueva población no dominada se excede, se aplica un método de reducción como lo es la reducción aplicando clustering.

**Figura 32. Función de no-dominancia**

```

Void Non_Dominated_Function (&New_Pop P')
For (All Solutions (i))
    For (Each level)
        If (Util(i)<Util(i+1)) Lose++;
        If (Util(i)>Util(i+1))Win++;
        If (Util(i)=Util(i+1))Tie++;
        If (Cost(i)<Cost(i+1))Win++;
        If (Cost(i)>Cost(i+1))Lose++;
        If (Cost(i)=Cost(i+1))Tie++;
    End For
    If (Win>0 and Lose=0)
        Dominated (i) // New_Pop
Else
    If (Win>0 and Lose>0)
        Both_ No Dominated // New_Pop
End if
End For
End Void

```

Fuente: Propia

#### 4.3.4.3 RUTINA DE REDUCCIÓN UTILIZANDO CLUSTERING

Como parámetro de la meta-heurística se tiene un número máximo de no dominados los cuales se pueden ajustar dependiendo del problema. De todas formas, dentro de los procesos que tiene el AGHMO para la evaluación de no dominados tiene una función de reducción por cluster, que consiste en agrupar las soluciones cercanas en clusters hasta que se tenga el número máximo de no dominados, si es que la nueva población sobrepasa este parámetro. La técnica de clustering utiliza las siguientes funciones

- Cálculo de distancia entre clusters
- Fusión de soluciones en clusters

- Cálculo de centroides por cluster

La función de clustering que utiliza el AGHMO para reducción de soluciones no dominadas, es la siguiente:

**Figura 33. Función de reducción aplicando clustering**

```

Void Clustering (New_ Pop P')
Clear_ Clusters ();
While (Num_New_Pop <= Max_ Non Dominated)
For (All Clusters)
Calc. Distance and select the two nearest a , b;
End For
Call Join_ Clusters (a,b);
Num_ New_ Pop--;
End While
For (All Clusters)
Run Calc_ Centroide_ Cluster ();
End For
End Void

```

Fuente: Propia

Una vez es calculado el centroide y se ha reducido la población al tamaño máximo permitido, se procede con las funciones de cálculo de fuerza y de fitness.

#### 4.3.4.4 FUNCIÓN PARA EL CÁLCULO DE STRENGHT Y FITNESS

La fuerza (Strenght) se calcula sobre la población no dominada ya que es la que lleva el peso sobre las otras soluciones. La forma de calcularlo es contar cuántas soluciones de la población normal domina la solución elitista, luego ese número se divide ente el número de la población más 1.

$$Strenght_i = n_i / (N + 1) \quad (4.1)$$

**Figura 34. Cálculo del Strenght y Fitness de población No-Dominada**

```
Void Calc_ Strenght (P')  
For (All Non Dominated Population (i))  
    For (All population dominated (j))  
        If (i dominates or ties to j)  
            Dom++;  
        End For  
    Strenght (i) = Dom/ (Population_ Dominated +1);  
    Fitness (i) = 1 / Strenght (i);  
End For  
End Void
```

Fuente: Propia

Se puede ver que el cálculo del fitness de las soluciones no dominadas depende de su propia fuerza sobre la población dominada. No pasa así con el fitness de la población dominada. Este fitness depende de la fuerza de las soluciones que dominan particularmente a cada solución de esta población, para garantizar poder distinguirla se le suma 1 a esta fuerza.

$$Fitness_i = 1 + \sum Strenght_i, i \in P' \quad (4.2)$$

Siendo P' la población no dominada resultante de la reducción por clustering. A continuación se presenta la función que calcula el fitness para toda la población dominada. De esta forma ya todas las soluciones tienen un fitness propio por medio del cual se pueden categorizar o priorizar.

**Figura 35. Calculo de fitness de población dominada**

```
Void Calc_ Fitness ()  
For (All Dominated Population (i))  
    For (All non dominated population (j))  
        If (j dominates to i)  
            Sum+=Strenght (j);  
        End If  
    End For  
    Fitness (j) = 1,0 + Sum;  
End For
```

**End Void**

Fuente: Propia

#### 4.3.4.5 MÉTODO DE SELECCIÓN

Una vez se tienen los fitness de las dos poblaciones, se realiza una selección estocástica sobre P y P'. El método empleado para el AGHMO fue el método de la ruleta. La ruleta toma el porcentaje acumulado que cada una de las soluciones tiene sobre el 100%. La selección se toma como base a un número aleatorio entre 0 y 1, el cual da el intervalo de probabilidades de las soluciones que serán seleccionadas. Por ejemplo si nuestro número aleatorio es 0.6, esto quiere decir que todas las soluciones que estén acumuladas hasta el 60% serán tomadas para la nueva población. Las mejores soluciones siempre son la base de la ruleta para que siempre se tomen las no dominadas, dando margen para que también se seleccionen unas dominadas que al momento de aplicar los operadores de evolución y optimización local, puedan dar una solución factible y muy buena. La función de selección aplicando el método ruleta es la presentada a continuación:

**Figura 36. Selección utilizando el método de la ruleta**

```
Void Roulette_Selection ()  
Get_Sum_Fit (); //Sum All Fitness of the populations P and P'  
Get_prob_sol (); // Obtain probabilities of the solutions. Sum of the prob = 1  
Sort_Prob (); // Sort all prob. Descendent.  
Rand1; // Prob: 0-1;  
Choose all solution between 0 and Rand1 prob. //New Pop Ps  
End Void
```

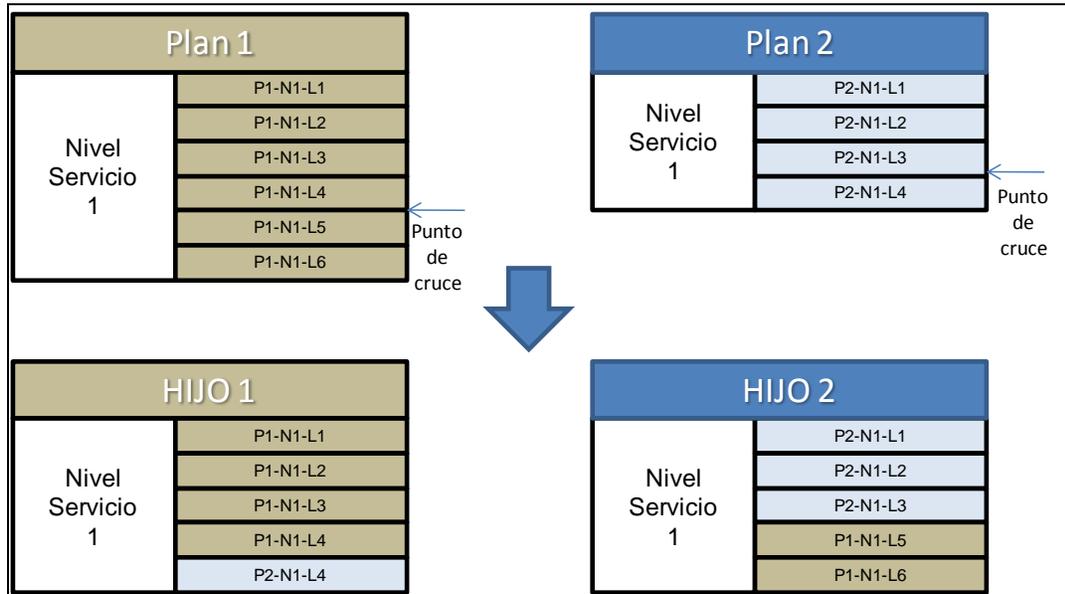
Fuente: Propia

Esta población Ps se obtuvo por medio de esta selección y será sometida a los operadores de evolución en busca del mejoramiento de estas.

#### 4.3.4.6 OPERADORES DE CRUZAMIENTOS

El operador de cruzamiento es el operador evolutivo que más importancia tiene dentro de una técnica como la estudiada, dado que la técnica de cruces ayuda a explorar todo el campo de acción de la solución sin estar enfocándose en óptimos locales. La idea de este operador es cruzar entre los diferentes planes seleccionados y niveles de servicio que los componen, las líneas para crear de esta forma otros planes hijos. Luego estos planes se hacen factibles de acuerdo a restricciones y se unen a la nueva población de cruce  $P_c$ . Esto se puede ver mejor en la figura 38.

**Figura 37. Estructura del cruzamiento general para el AGHMO**



Fuente: Propia

Esta función depende de un parámetro probabilístico que indica la frecuencia de ejecución del operador. Ese parámetro es la probabilidad de cruzamiento. La literatura especializada recomienda, que por la importancia de este operador, este parámetro este entre 80% y 95%. Para algunas soluciones se ha probado el 60%.

**Figura 38. Función para el operador de cruzamientos**

```
Void Cross_ Over_ Function (Cross_ Prob)
For (All options in the new pop (i))
  Rand1; //Random Number between 0 – 1
  If (Rand1<=Cross_ Prob)
    Cross (i) with (i+1);
    Set New_ Pop_ Cross ((i), (i+1)); //Assign new solution to Pc
  End If
End For
End Void
```

Fuente: Propia

Estos cruces se hacen sobre Ps y es ejecutada con la esperanza de que Ps, que consta de partes buenas de la población anterior, sea mejor. Sin embargo, es bueno dejar que alguna parte de la población P anterior sobreviva a la siguiente generación.

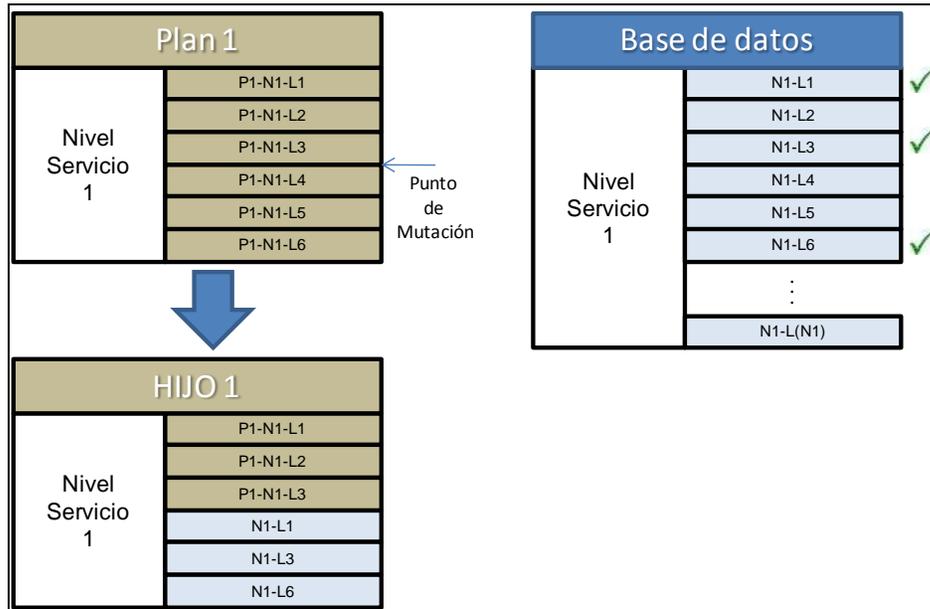
#### **4.3.4.7 OPERADORES DE MUTACIÓN**

El operador de mutación es el encargado de crear una solución mediante la modificación de una existente, incluyendo conocimiento adquirido generalmente acerca del problema. Este operador se aplica sobre la población Ps, obteniendo así la población Pm, la cual se unirá a la población obtenida por cruzamientos Pc para crear una nueva.

Este operador toma cada opción de plan seleccionado y lo muta, partiendo cada nivel del plan seleccionado y completándolo con líneas que hacen factible el plan para ese nivel específico.

La mutación se hace en dos niveles: a nivel de los planes y sus niveles, y a nivel de las líneas que componen cada nivel. A nivel de las líneas, están se parten y se reconstruyen con agencias que se toman de forma aleatoria da tal manera que sea factible de ir de una agencia hasta la otra.

**Figura 39. Estructura del operador de mutación general**



Fuente: Propia.

**Figura 40. Función para el operador de mutación**

```

Void Mutation_ Function (Mutation_ Prob)
For (All option selected (i))
  Rand1; // Random number between 0-1
  If (Rand<=Mutation_ Prob)
    Mutate option (i) and option's lines
    Set New_ Pop_ mutated (i) // Assign solution to the population mutated
  End If
End For
End Void

```

Fuente: Propia

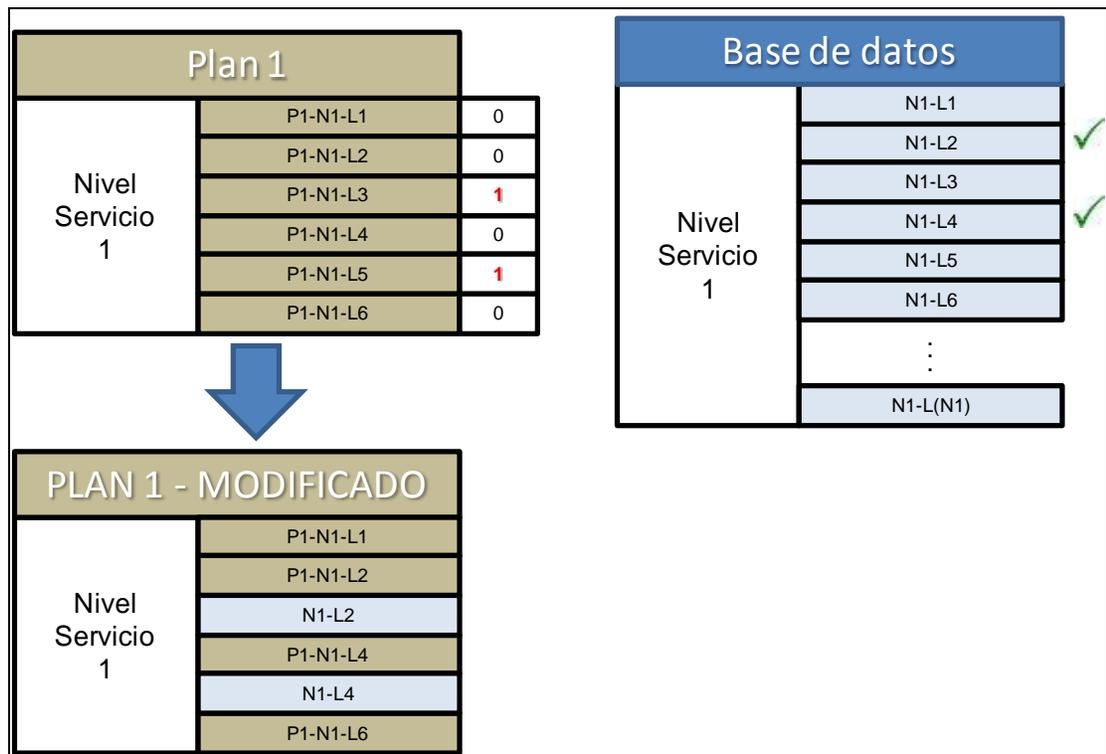
Una vez ha sido aplicada la función de mutación y cruzamiento, se aplica sobre Pc y Pm, de manera aleatoria, el meta-operador de optimización local.

#### 4.3.4.8 META-OPERADOR DE OPTIMIZACION LOCAL

Estos meta - operadores son uno de los rasgos más distintivos de los AGH. Ellos iteran la aplicación del operador de mutación y cruzamiento, conservando los rasgos del que obtuvo alguna mejora en la creación de la nueva solución.

El funcionamiento es muy sencillo. Se escoge de forma aleatoria soluciones de la población cruzada  $P_c$  y de la población mutada  $P_m$ . A estas soluciones se les aplica una máscara binaria, la cual mostrara que líneas deben ser mutadas o cambiadas.

**Figura 41. Estructura de funcionamiento del optimizador local**



Fuente: Propia

Se puede ver claramente en la figura anterior como un plan seleccionado fue transformado aplicando una máscara binaria que se asigna de forma aleatoria, donde 1, es la línea que debe ser transformada o cambiada. Este proceso no crea

una población diferente, lo que hace es transformar un plan y luego evalúa si este domina al original para reemplazarlo en la población a la cual pertenece.

**Figura 42. Función de optimización local**

```
Void Local_Optimization(Pc,Pm)
For (All Pc and Pm -> i)//Randomly
Apply Mask mutation for solution i (new_solution j)
    If(new_solution(j) dominates i)
        Repleace i for j
    End if
End For
End Void
```

Fuente: Propia

Una vez aplicados los operadores evolutivos, se procede a actualizar la población P para evaluar la próxima generación. Esta función toma las soluciones no dominadas P', las soluciones que se obtuvieron por cruzamiento Pc, mutación Pm después de la aplicación del meta-operador de optimización local. Esto se hace para cada generación permitiendo la evaluación de las nuevas soluciones con las mejores de cada generación.

#### **4.3.5 VENTAJAS Y DESVENTAJAS DEL MODELO AGHMO**

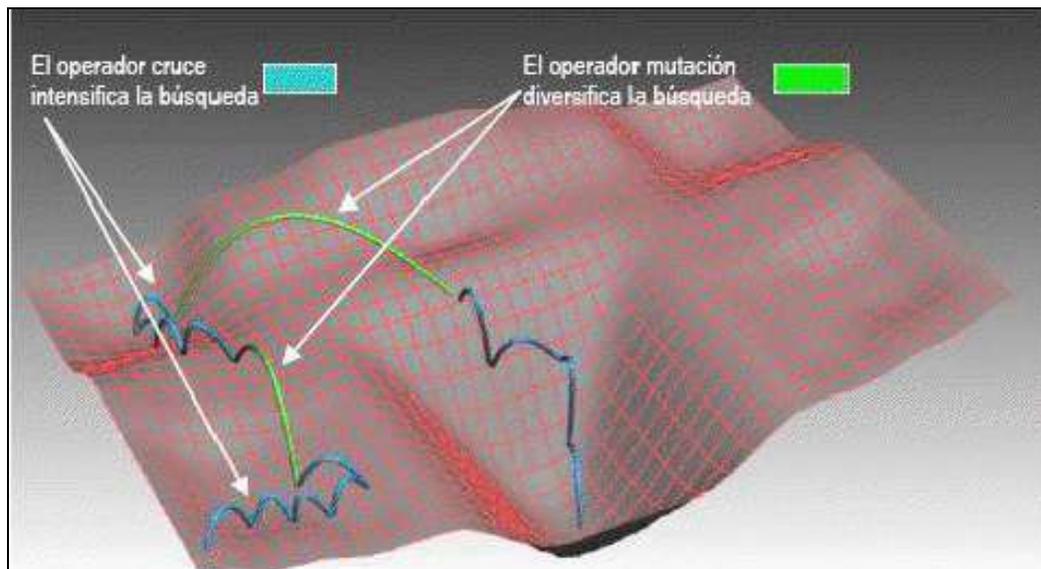
##### **4.3.9.1 VENTAJAS**

Una de las ventajas más grandes que tiene el AGHMO planteado, es que su base es totalmente evolutiva, es más, está su cuerpo en una gran parte, se basa en el SPEA (por sus siglas en ingles, Strenght Pareto Evolutionary Algorithm), y eso lo hace trabajar en paralelo sobre un conjunto de soluciones que tienen la potencialidad de tratar problemas con múltiples objetivos, hallando en cada ejecución un gran conjunto de soluciones aproximadas al frente de Pareto óptimo. El algoritmo es menos sensible a la forma o a la continuidad del frente de Pareto,

ya que el grado de escalabilidad del modelo permite la evaluación de zonas en el espacio de solución bastante distantes con el fin de encontrar óptimos globales.

Además el AGHMO, permite que se aborden problemas de grandes espacios de soluciones y tienen una gran habilidad para optimizar varios parámetros simultáneamente. Por lo general, los problemas más complicados no son definibles en función de un parámetro si no que son necesarios varios para poder dar una solución válida. De igual forma las relaciones entre estos parámetros suelen ser inversas, es decir, para mejorar uno debemos empeorar otros. Haciendo uso del paralelismo anteriormente mencionado, el AGHMO es capaz de encontrar múltiples soluciones para cada parámetro, siendo todas ellas igual de válida y factibles.

**Figura 43. Variaciones en el espacio de búsqueda de un AGHMO**



Fuente: Abascal Pelayo, Feijoo Ugalde<sup>27</sup>

Para finalizar esta apartado de ventajas haremos uso de la descripción que hace

<sup>27</sup> Abascal Pelayo, Feijoo Ugalde. 2009. Implementación de Algoritmos Genéticos sobre la plataforma de desarrollo paralelo CUDA. Universidad Complutense de Madrid

*Dawkins*<sup>28</sup> de los algoritmos genéticos híbridos, “*relojeros ciegos*”. Esta expresión hace hincapié en el concepto de que un AGH es totalmente independiente del problema que esté tratando en cada momento. Al no necesitar un conocimiento previo como otros algoritmos (ya que basan toda su potencia de trabajo en la aleatoriedad), los AGH no descartan ningún camino a priori, manteniendo de esta forma la posibilidad de encontrar soluciones novedosas que no se les haya ocurrido a los expertos.

#### **4.3.9.2 LIMITACIONES**

Antes tan siquiera de pensar en el AGH en sí, debemos plantearnos la representación por la que vamos a optar para dar forma al espacio de búsqueda. Esta representación debe de ser lo suficientemente sólida como para aceptar cambios aleatorios sin llevar a producir errores ni individuos sin sentido.

La función de ajuste es otro gran reto a la hora de implantar el AGHMO. Ésta debe cumplir los requisitos que hagan que realmente encuentre la mejor solución sin llegar a recortar la aptitud de ninguna solución. Recordemos que al no tener el frente de Pareto verdadero como solución del problema, no podemos medir métricamente que tan buena es o no la solución. Pero más que un defecto del modelo, es del problema en sí. Necesitaríamos de un método exacto que nos dé el resultado del frente real.

Además de esto, también los parámetros del algoritmo (tamaño de la población, población elitista, tipo de selección, probabilidad de mutación y cruce, mascara binaria para la optimización local) deben ser elegidos con cautela. El tamaño de la población incide directamente sobre la cantidad de soluciones que el algoritmo explore, si este es bajo el algoritmo no trabajará lo suficiente y probablemente no encuentre soluciones muy buenas, en cambio si es muy grande deberemos

---

<sup>28</sup> Dawkins, Richard. *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe Without Design*. W.W. Norton, 1996

sacrificar tiempo de cálculo. Si los operadores de cambio genético operan demasiado, puede dejar de buscar cerca de soluciones prometedoras ya que no convergería lo suficientemente rápido para el ritmo de cambio impuesto.

Otro de los problemas, del que ya hemos hecho mención anteriormente, es el de la convergencia prematura. Si en las primeras etapas del algoritmo la población dispone de una solución netamente superior al resto (con un fitness mucho mejor), ésta se reproducirá rápidamente haciendo que el AGHMO converja, limitando la diversidad y en consecuencia las posibles soluciones. Las poblaciones con un bajo número de soluciones son las más proclives a este tipo de problemas, ya que la más mínima variación provoca que un individuo llegue a dominar al resto. El abanico de soluciones que los investigadores han encontrado al problema de la convergencia prematura pasan por restringir la selección, para no dar tanta ventaja a los individuos excesivamente aptos. La selección *escalada*, *por rango* y *por torneo*<sup>29</sup>, son tres de los métodos fundamentales para lograrlo.

---

<sup>29</sup> Abascal Pelayo, Feijoo Ugalde. 2009. Implementación de Algoritmos Genéticos sobre la plataforma de desarrollo paralelo CUDA. Universidad Complutense de Madrid. Página 23.

## **5. CASO DE APLICACIÓN: DSS-BRASILIA SISTEMA DE SOPORTE DE DECISIONES EN TRANSPORTE TERRESTRE DE PASAJEROS.**

### **5.1 DESCRIPCIÓN**

En este capítulo se describe la aplicación de la metaheurística AGHMO en un caso de uso real en una empresa del sector transportador terrestre de pasajeros EXPRESO BRASILIA S.A. ubicada en la ciudad de Barranquilla, Colombia. Se explicará el problema que se quería resolver y cómo fue adaptado el AGHMO en un sistema de soporte de decisiones de la empresa (DSS-BRASILIA). Al final del capítulo se mostraran los resultados hallados.

### **5.2 EXPLICACIÓN PROBLEMA A RESOLVER**

Expreso Brasilia S.A. es una empresa del sector transportador terrestre de pasajeros en Colombia, específicamente sus operaciones se centran en la parte norte del país. Esta empresa lleva muchos años de operación, y con el tiempo ha visto la necesidad, dadas las condiciones de la competencia entre otras, poder hacer eficaz sus operaciones alrededor del país.

Con el tiempo, ellos han creado una base solida de diferentes líneas y rutas, las cuales se encuentran compuestas por un número fijo de agencias, las cuales cumplen características especiales dadas por restricciones gubernamentales o de la misma compañía.

**Figura 44. Mapa de operaciones de Expreso Brasilia S.A.**



Fuente: <http://www.expresobrasilia.com/espanol/agenciasyterminales.html>

La empresa cuenta con una serie de niveles de servicio especializados en diferentes tipos de líneas y sectores del país. En el momento de la evaluación, la empresa cuenta con 6 niveles de servicio jerárquicamente puestos así:

- Nivel Preferencial internacional
- Nivel Preferencial Nacional
- Nivel Gaviota 96-98
- Nivel Gaviota 93-95
- Nivel Coche Baño
- Nivel COM

Cada nivel de servicio tiene asignado un número importante de líneas, que permiten la cobertura de operaciones de la empresa.

Teniendo esto en cuenta, el problema que estudió la empresa fue cómo desarrollar un sistema de soporte para la toma de decisiones (DSS) que permita la planificación estratégica de sus operaciones de transporte terrestre de pasajeros, y que sea integrado fácilmente a su sistema actual de gestión empresarial, con el fin que la empresa pueda administrar en forma integrada y efectiva los recursos de transporte con decisiones optimizadas de asignación de rutas, conductores y flota vehicular, y facilite el establecimiento de tarifas de operación.

Específicamente en el problema de planeación estratégica de las operaciones de transporte, se propuso como caso de optimización multiobjetivo, aplicar el AGHMO. Los objetivos a optimizar planteados por la empresa fueron los siguientes:

- Minimización del costo anticipo de cada nivel de servicio
- Maximización de la utilidad por nivel de servicio
- Minimización de la varianza de ingresos entre buses de un mismo nivel de servicio.

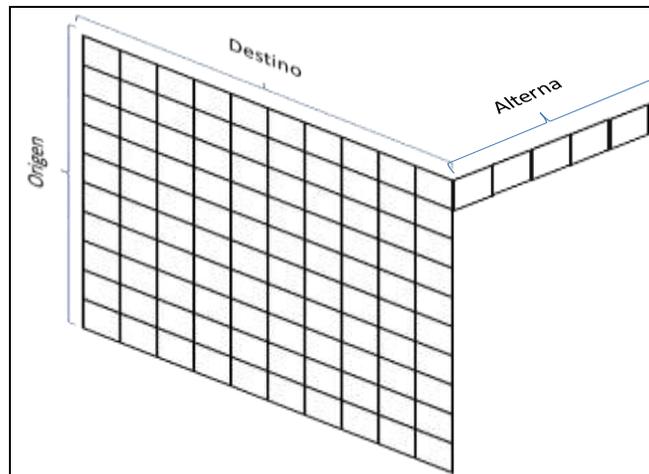
El AGHMO fue adaptado e integrado con éxito para la optimización empleando los dos primeros objetivos. El tercer objetivo fue diseñado a manera de restricción del modelo y no se describe en el presente trabajo. A continuación veremos los cambios específicos se hicieron en el AGHMO para ser incorporado con éxito en el sistema para el soporte de decisiones de Expreso Brasilia S.A (DSS-Brasilia).

### **5.3 RESTRICCIONES, FUNCIONES Y VARIABLES ESPECÍFICAS DEL PROBLEMA.**

Una solución factible para Expreso Brasilia debe contar con una serie de restricciones propias de la empresa y del gobierno Colombiano. Veamos específicamente cada una de las variables y valores de parámetros que fueron usados para el acople del AGHMO al DSS-Brasilia:

- **Rutas:** En Brasilia existen varias firmas de ir desde una agencia A hasta una agencia B. Ellos dan el nombre de rutas alternas a toda ruta que me permita llevar desde una agencia A hacia un B y que no se la principal definida por ellos. Este tipo de rutas son tomadas en el modelo, dado que es posible que las rutas principales estén cerradas en el tiempo de ejecución. Esto hace que la matriz origen destino no sea bidimensional, sino que se plantee tridimensional.

**Figura 45. Estructura de matriz origen, destino, alterna**



Fuente: Propia

- **Horarios:** los horarios serán manejados como múltiplos de 5 minutos, lo que significa períodos discretos de 5 minutos. Si según fue validado con la Gerencia de Operaciones es suficiente este nivel de precisión para la planeación y programación de las operaciones. Es importante aclarar que el periodo más corto de planificación será de tres meses, 90 días. El modelo funcionara igual sin importar la cantidad de tiempo que incluyan como parámetro.
- **Buses Disponibles Por Nivel De Servicio:** Para cada ejecución del AGHMO debe quedar un numero buses disponibles igual a lo menos el 20% de los buses activos.

- **Tiempos De Descanso Entre Llegadas Y Salidas:** Se conocen diferentes tipos de tiempo (alistamiento, viaje, paradas intermedias, de descanso entre paradas y salidas) pero estos valores serán incluidos de manera agregada como parámetros del sistema.
- **Tiempo Máximo Para Despacho:** Indica el tiempo máximo en que se puede permanecer en rampa para un próximo lanzamiento.
- **Requerimiento De Conductor Relevo Ruta-Horario:** Se trata de un parámetro que define las rutas-horarios en las que es necesario incluir un conductor de relevo. Debe aplicarse a todas las rutas de recorrido de más de 10 horas continuas. Ojo, usar relevo en tramo que se necesita.
- **Horario De Funcionamiento Oficinas:** Se trata de ventanas de tiempo para la llegada a las agencias. Si cierra la oficina no recoge más gente, sólo es para dejar. Afecta la productividad de la ruta.

Veamos cómo se ajustaron los parámetros propios de la metaheurística:

- **Numero de soluciones iniciales:** Es la cantidad de soluciones que pasarán por la primera generación del AGHMO. Este valor es parametrizable, para que sea cambiado conforme a la decisión del experto. El parámetro no debe ser menor de 10 soluciones, para que no se limite el accionar del modelo.
- **Numero de soluciones elitistas:** Es el valor máximo del conjunto de soluciones P' que dominan a las otras. Este valor es parametrizable, para que sea cambiado conforme a la decisión del experto. Se establecieron máximo 5 soluciones elitistas.
- **Probabilidad de Cruzamiento:** Este valor define la frecuencia con que se ejecutará el operador de cruzamientos. La probabilidad de ejecución fue de una 80%. Este valor es parametrizable también, sin poder disminuir menos de un 60% y sin llegar a más de un 95%. Estos parámetros fueron obtenidos a través de un diseño experimental sobre el modelo, que mostro cual porcentaje mejoraba las soluciones. El resultado, apoya la literatura especializada sobre asignar el 80% de la probabilidad al operador de cruzamiento.

- **Probabilidad de Mutación:** Este valor define la frecuencia con que se ejecutará el operador de mutación. La probabilidad seleccionada de operación fue 15%. Este valor es parametrizable, sin poder disminuir menos de un 10% y sin llegar a más de un 20%. Este parámetro también fue seleccionado conjunto al anterior en el mismo diseño experimental.

**Figura 46. Parámetros de operación del AGHMO desde el DSS-Brasilia**

PARAMETROS INICIALES	
DIA INICIO RODAMIENTO	DIA FINAL RODAMIENTO
30/08/2006	30/08/2006
<b>Parámetros de Optimización</b>	
Numero maximo de generaciones	3
Numero maximo de No-Dominados	5
Probabilidad de cruzamiento de planes	80
Probabilidad de mutacion de planes	15
<input type="checkbox"/> Restringir planes - Utilidad por Bus	
<b>Parámetros de planes hijos</b>	
Tiempo minimo entre rutas:	1,5
Tiempo minimo entre lineas:	8,0
Tiempo maximo en movimiento:	20,0
Tiempo descanso conductor:	8,0
Tiempo descanso Intermedias:	0,25
<input type="button" value="ACEPTAR"/> <input type="checkbox"/> Calculo Tiempos-Ruta <input type="checkbox"/> Calculo Peajes-Ruta	

Fuente: DSS-BRASILIA

### 5.3.1 MODIFICACIONES PARA IMPLEMENTACIÓN DEL AGHMO EN DSS-BRASILIA

Para la adaptación del AGHMO al problema de planificación de Expreso Brasilia S.A. no se encontraron grandes problemas, más bien fue necesario modificar ciertas funciones por la naturaleza del problema de la empresa.

Una de esas modificaciones, se hicieron en la función para eliminar cruces entre las diferentes rutas. El objetivo de esta función, es evaluar y eliminar todos los posibles cruzamientos que se puedan presentar en las agencias y en el camino una vez se hayan asignado los tiempos de salida y llegada. Esto se hace por diferentes políticas que plantea la empresa, entre ellas evitar tener un alto uso de los terminales en un momento dado; evitar también los roces y peleas en el camino.

Teniendo en cuenta parámetros que son administrados desde tablas en la base de datos se sabe cuánto tiempo deben estar desfasados los diferentes vehículos, si:

- Utilizan la misma agencia en un instante de tiempo para hacer trasbordos o dejar pasajeros o ya sea para cambiar de pasajeros.
- Utilizan vías intermedias o alternas para llegar a las diferentes agencias.

Esta función evalúa todos los niveles de servicio entre ellos dado que es posible tener cruces entre los vehículos de los diferentes niveles. La función da prioridad a los niveles de mayor importancia para la empresa. La jerarquía se evalúa como sigue: 1) Preferencial Internacional, 2) Preferencial Nacional, 3) Gaviota 96-98, 4) Gaviota 93-96, 5) Coche baño.

En este momento solo existen 2 centros operativos: Barranquilla y Medellín. La jerarquía anterior presenta los niveles de servicio para el centro operativo de Barranquilla. Para el centro operativo Medellín sólo existe uno en este momento llamado COM, el también evalúa cruces con los demás niveles pero no entra dentro de la jerarquización anterior dado que es un centro operativo aparte.

Esta función se reemplaza en la generación de planes semillas y en la parte que se hace factible cada solución.

**Figura 47. Función para eliminación de cruces entre agencias DSS-Brasilia**

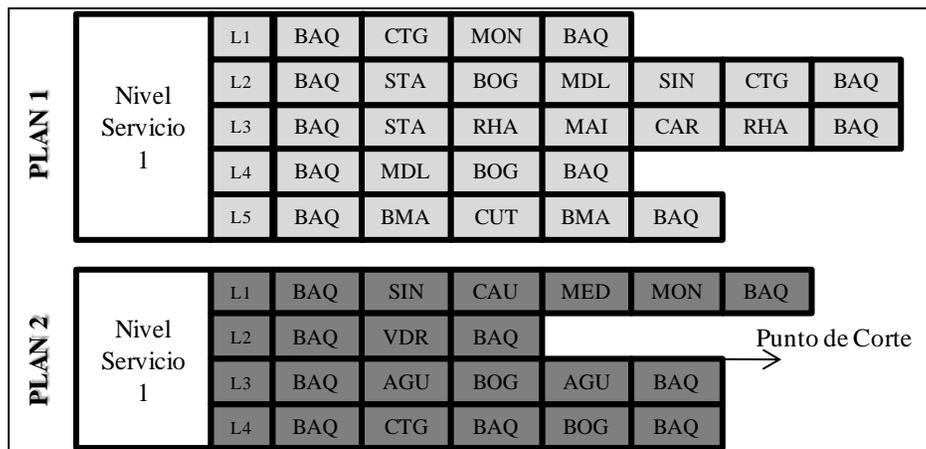
```

Void Crossover_ Function ()
For (All level until Present)
    If (Level is less that another)
        For (All Lines in Level)
            While (! Each O-D of the line has been evaluated)
                Times Routes of the present level are adjusted depending of
                the parameters in the Data bases.
            End While
        End For
    End If
End For
End Void
    
```

Fuente: Propia

Por otro lado, se añadieron cambios a los operadores evolutivos de mutación y cruzamiento. El modelo inicial actúa eligiendo puntos de quiebre dentro de los niveles de servicios para estos ser cruzados con otros o mutados con más líneas. En la implementación se desarrollaron dos operadores que trabajan desde el punto de vista de la línea. No se eliminaron los operadores que se habían explicado, solo se adicono uno para trabajar nivel de líneas y no de niveles de servicio.

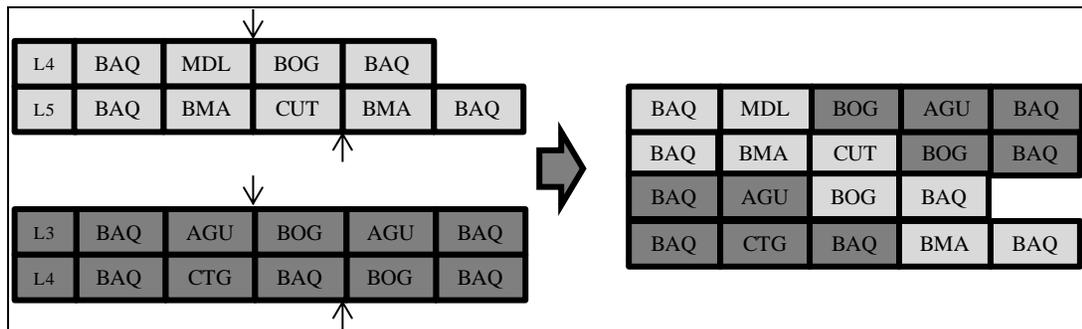
**Figura 48. Funcionamiento del operador de cruzamiento a nivel de la línea.**



Fuente: Propia

El operador de cruzamiento, toma dos planes padres de la población elitista o no dominada, y luego dos niveles de servicio idénticos dentro de cada plan. No importa cuantas líneas tenga cada uno. Se hace un corte aleatorio tomando como base el plan que tenga menos líneas en ese nivel. La función opera de manera probabilística cruzando las líneas de un plan con las del otro. Como primera medida se selecciona un punto de corte dentro de cada nivel de los planes seleccionados. Este punto será escogido de forma aleatoria tomando como referencia el nivel que tenga menos cantidad de líneas. Luego, se escoge un número de líneas iguales del punto de corte hacia el final del plan para ese nivel. Por ejemplo, si el nivel que tiene menos líneas desde el punto de corte hasta el final cuenta con 2 líneas, se tomaran las dos líneas finales de cada nivel para cruzarlas.

**Figura 49. Resultado del cruce de líneas**



Fuente: Propia

Después de cruzar las líneas, los planes hijos resultantes quedarían conformados, por las líneas desde el punto de corte hasta el inicio del plan inicial agregando, aleatoriamente las líneas resultantes por cruce a cada plan como se muestra en la siguiente figura.

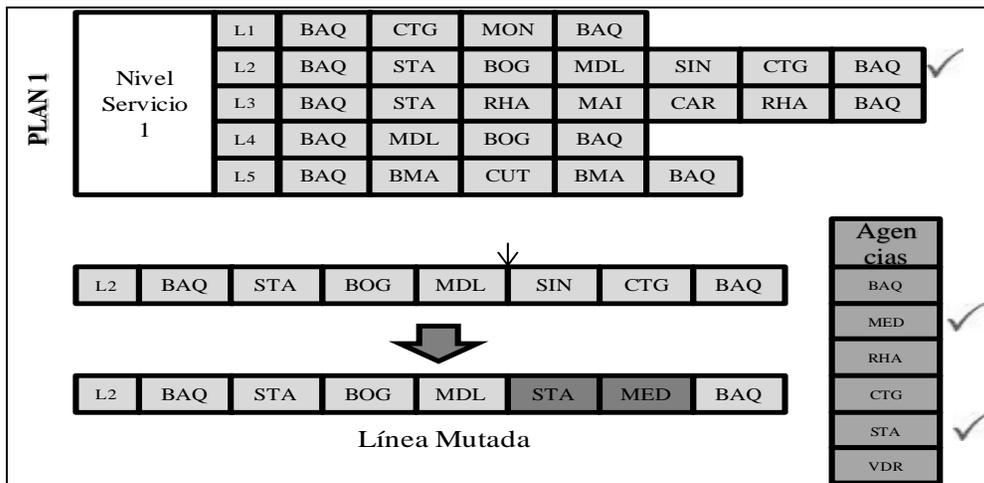
**Figura 50. Planes hijos producto del cruce entre líneas**

<b>HJO 1</b>	Nivel Servicio 1	L1	BAQ	CTG	MON	BAQ			
		L2	BAQ	STA	BOG	MDL	SIN	CTG	BAQ
		L3	BAQ	STA	RHA	MAI	CAR	RHA	BAQ
		L4	BAQ	MDL	BOG	AGU	BAQ		
		L5	BAQ	CTG	BAQ	BMA	BAQ		
<b>HJO 2</b>	Nivel Servicio 1	L1	BAQ	SIN	CAU	MED	MON	BAQ	
		L2	BAQ	VDR	BAQ				
		L3	BAQ	BMA	CUT	BOG	BAQ		
		L4	BAQ	AGU	BOG	BAQ			

Fuente: Propia

Ahora veamos el operador de mutación. Este elige de forma aleatoria un plan, sea de Ps o Pm. La idea es crear un nuevo plan, simplemente mutando ciertas líneas del plan existente. Las líneas se seleccionan de forma aleatoria para cada nivel de servicio. Una vez se escoge una línea esta se corta y se escoge un numero aleatorio no mayor a la extensión de la línea, esto indica la cantidad de agencias que se añadirán desde el punto de corte. Estas agencias son escogidas desde la base de datos.

**Figura 51. Selección de una línea y punto de corte para mutación**



Fuente: Propia

Se pueden implementar muchas formas de aplicar cualquiera de estos operadores, unas más sencillas que otras. El punto es que el AGHMO mostro muy buenos resultados en corto tiempo aplicando estas funciones y operadores como lo hemos descrito.

### **5.3 EVALUACIÓN Y RESULTADOS**

El desarrollo aplicado en el DSS-Brasilia utilizando el AGHMO, mostro grandes resultados en tiempos realmente accesibles. El AGHMO fue probado en varios escenarios, pero realmente solo nos interesaba saber el comportamiento en un escenario de operación continuo de 3 meses (90 días por petición de la empresa).

La aplicación que se construyo bajo el lenguaje de programación C++, tomó un tiempo promedio de 12 minutos para evaluar en 10 generaciones más de 100 posibles soluciones. Al final de las generaciones se presenta aquellas que por reducción son las más acertadas y cercanas al óptimo. Cada una de estas soluciones podría ser implementada por la empresa en tiempo de operación continuo de 3 meses.

Los resultados que se tomaron como referencia, fue La operación que la compañía tuvo en el mes de agosto del año 2009. Los datos arrojados fueron los siguientes, cabe aclarar que dichos resultados fueron dados por la compañía y apoyan la publicación de estos:

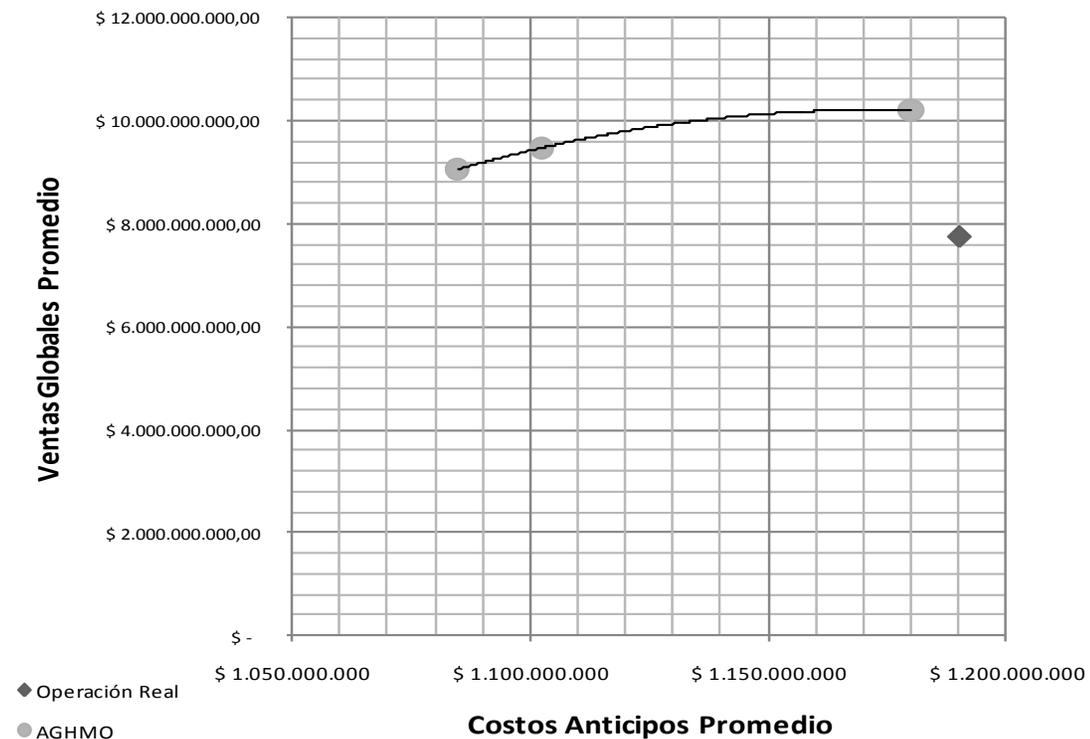
**Figura 52. Tabla de comparativa de resultados del AGHMO vs Operacion Real**

	Costo Anticipo Promedio	Ventas Globales Promedio	Utilidad Promedio
Operación Real	\$ 1.190.015.862,00	\$ 7.741.481.684,80	\$ 6.551.465.822,80
AGHMO 1	\$ 1.180.069.000,00	\$ 10.198.982.656,00	\$ 9.018.913.656,00
AGHMO 2	\$ 1.102.402.900,00	\$ 9.473.349.780,00	\$ 8.370.946.880,00
AGHMO 3	\$ 1.084.812.000,00	\$ 9.056.457.240,00	\$7.971.645.240,00

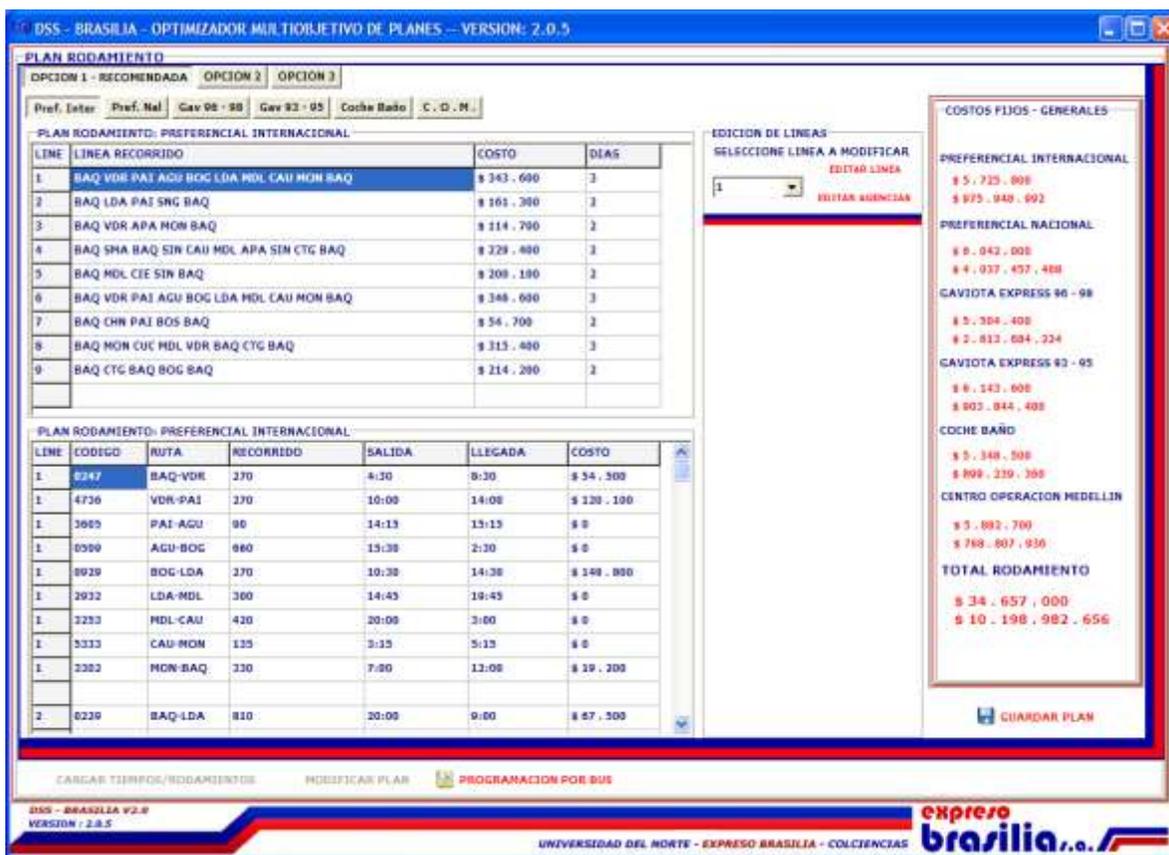
Fuente: Resultados DSS-Brasilia después de la ejecución del AGHMO

Se puede notar que existen 3 resultados del AGHMO. Cabe recordar que el AGHMO no arroja una solución, sino que da un conjunto de soluciones óptimas que entre si son No-Dominadas o no comparables. Es se puede ver mejor en la figura a continuación, como las soluciones representan un frente de Pareto aproximado.

**Figura 53. Frente Pareto descrito por el AGHMO**



**Figura 54. Reporte del DSS-Brasilia aplicando el AGHMO**



Fuente: DSS-Brasilia

Adicional al logro alcanzado a implementar el AGHMO en el DSS-Brasilia obteniendo buenos resultados en excelentes tiempos, la empresa logro estandarizar los registros y optimizó el diseño que tenían de su base de datos, motivados por los requerimientos de información confiable, para las consultas informáticas de la herramienta. Anterior al proyecto, la información existente o los procesos de ingreso y codificación presentaban un alto porcentaje de incongruencias. La estandarización permitió obtener resultados confiables y procesar la información de una manera más estructurada y ágil.

El principal y más importante efecto del proyecto, fue la optimización bajo un elevado grado de, restricciones operativas y la efectiva consecución de planes de

operación de los vehículos que componen la flota de la empresa. Este logro redujo los tiempos de análisis para la generación de planes de rodamientos, de 1 mes, a una fracción de 1 día, permitiendo llegar a soluciones de mayor calidad que las desarrolladas por la única vía del experto.

También ayudo a la incorporación de las restricciones operativas, que obligan al sistema a desarrollar planes que garanticen el cumplimiento de las normas operativas de la compañía y las impuestas por el gobierno. También establece equilibrio entre el promedio de ingreso por bus, lo cual permite una mejora considerable en cuanto a la eliminación de preferencias sobre grupos particulares de conductores o vehículos. El desarrollo logra también la reducción del número de eventos de cruzamientos de vehículos en tramos, lo cual evita la probabilidad de conflictos por competencias de rutas comerciales.

Estos resultados comentados fueron apoyados por un sistema dinámico que simuló los planes de ruteo dados para la empresa. Esta simulación hecha en el programa ARENA, podía simular un evento el cual daría como resultado la reconstrucción o reacción del plan dado por el AGHMO. Esta fase del proyecto no es el punto de interés de este trabajo, pero es importante resaltarla ya que se puede tener un desarrollo posterior utilizando el AGHMO en modelos reactivos. La fase de evaluación conto con el apoyo de los expertos de Expreso Brasilia S.A en darnos la confiabilidad de los datos y los resultados que al final se lograron.

## **6. CONCLUSIONES Y DESARROLLOS FUTUROS**

A continuación se muestran las conclusiones de este trabajo y su aplicación al caso de real en la empresa de transporte terrestre de pasajeros Expreso Brasilia S.A. También se plantean ciertos desarrollos y trabajos futuros que se pueden hacer tomando como referencia el trabajo realizado con el modelo del AGHMO planteado.

### **6.1 CONCLUSIONES**

En este trabajo se estudia el problema de optimización multiobjetivo de planes de ruteo, frecuencias y asignación de horarios de despachos y minimización de cruces en la red vial, en sistemas de transporte terrestre de pasajeros, utilizando un enfoque de optimización combinatoria. Se reseñan los principales modelos y algoritmos existentes para la resolución del problema.

Se propone un modelo de optimización multiobjetivo y un algoritmo aproximado para su resolución, basado en la metaheurística AGH (Algoritmos Genéticos Híbridos). La metodología propuesta es aplicada a un caso de desarrollo relativo a una empresa transportadora de pasajeros, Expreso Brasilia S.A. en Colombia, para cuya construcción fue necesario el diseño y ejecución de un buen sistema informático para la toma de decisiones DSS.

En cuanto a la resolución del problema, la metodología de esta tesis ha sido el tratar por separado dos aspectos que tienen relación estrecha entre ambos, y que son el modelado matemático y la resolución algorítmica.

Se logró validar el modelo del AGHMO en un caso de uso específico, y junto con expertos de la materia (de la empresa Expreso Brasilia S.A), se pudo adaptar el modelo y guiarlo a tiempos de ejecución excelentes para el tiempo de planeación de cada solución entregada por el sistema. Inclusive se tomaron planes históricos y se evaluaron contra lo real, lo cual permitió ver que tan próximos estábamos en la construcción de planes factibles.

Una de las grandes conclusiones sacadas de esta aplicación fue, que el AGHMO planifica de forma apropiada y de acuerdo a las restricciones (sin ser flexible), planes que permiten sobre pesar los objetivos propuestos por la compañía. Es más, el sistema en que se aplico, entrega 3 soluciones de las cuales ninguna domina a la otra y se puede aplicar. Uno de los grandes tropiezos encontrados era, justamente, a que grado la empresa podía cumplir este plan, dado que hay factores humanos y geológicos que no se pueden controlar, y se necesitaría que el modelo pudiese reaccionar tomando en cuenta dichos eventos, los cuales no aplican a este trabajo.

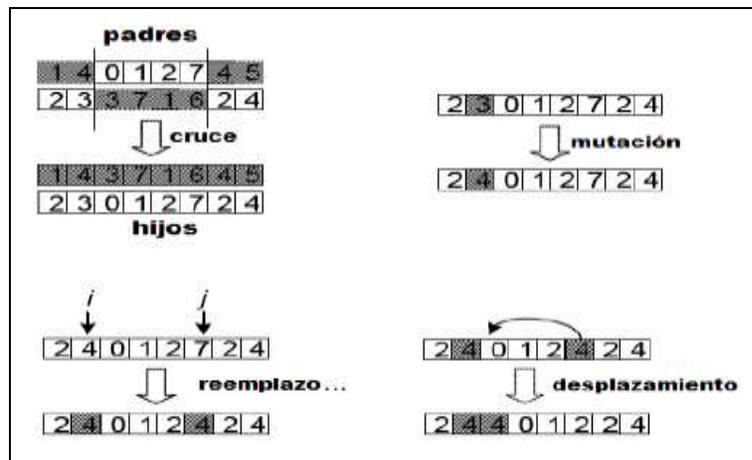
## **6.2 DESARROLLOS Y TRABAJOS FUTUROS**

Se identificaron las siguientes líneas de trabajo a seguir:

- *Formulación del modelo de optimización matemático.* La formulación propuesta para el modelo de optimización está fuertemente basada en formulaciones existentes y el apoyo bibliográfico que ésta tiene. La misma permite representar todas las características deseables en un modelo de optimización multiobjetivo de recorridos, frecuencias y asignaciones de despacho. Sin embargo, esta no es una formulación estricta de programación matemática. El

- interés de contar con una formulación de tales características puede permitir:
- 1) contar con una estructura reconocible que permita caracterizar el problema (el problema es NP-Hard por sus características, pero no existe un modelo que así lo confirme), 2) derivar un algoritmo exacto para la resolución del problema.
- *Algoritmo propuesto.* El algoritmo propuesto ha producido buenos resultados, ya sea frente a otros algoritmos existentes o con el caso de estudio de Expreso Brasilia S.A (DSS-Brasilia). Sin embargo se identifican las siguientes posibles mejoras en su diseño, que merecen exploración:
    - Mejora del optimizador local aplicando el operador EMMRS. es un operador genético que aúna en un solo paso las técnicas del crossover y de la mutación. Está basado en el operador genético establece que la reproducción entre cada par de individuos se realiza generando dos nuevos hijos con la información genética de los padres, enmascarada a través de un cromosoma, por cada individuo, destinado a tal efecto. Dicha información se recombina a través de dos puntos de cruce e introduce una mutación aleatoria en uno de los genes.

**Figura 55. Operadores genéticos del EMMRS**



Fuente: *Abascal y Pelayo*<sup>30</sup>. Página 39

<sup>30</sup> Abascal Pelayo, Feijoo Ugalde. 2009. Implementación de Algoritmos Genéticos sobre la plataforma de desarrollo paralelo CUDA. Universidad Complutense de Madrid. Página 39

- *Evolución Diferencial ED*: es otro algoritmo de búsqueda introducido más recientemente y que fue concebido inicialmente para la resolución del problema de ajuste de polinomios de Chebychev. Desde entonces, ha sido satisfactoriamente aplicado a un amplio espectro de problemas de optimización continua. Comparte con los AGH su naturaleza estocástica y su funcionamiento basado en poblaciones de soluciones. Difiere, sin embargo, en que, en el caso de la ED, los nuevos individuos son generados mediante un proceso de mutación, mientras que en los AGH el mecanismo principal de reproducción era el operador de cruce. El operador de mutación de la ED añade la diferencia ponderada de dos individuos seleccionados aleatoriamente a un tercer individuo para generar la nueva solución. Posteriormente, esta solución es combinada mediante un operador de cruce con un individuo objetivo para obtener una potencial solución. De entre estas dos soluciones, la obtenida por medio de los operadores de cruce y mutación y el individuo objetivo, se selecciona la de mejor fitness como representante para la siguiente generación.
- *Modelo reactivo del AGHMO*. El algoritmo propuesto, como ya lo hemos visto antes, no reacciona a eventos que se puedan presentar en cualquier instante de tiempo. La forma de hacer reactivo el plan, fue implementando un modelo de simulación que ejecutase el plan de ruteo seleccionado y en la marcha este modificase las variables que cambiaron en el entorno del plan. El mayor problema que se presentaba es que no se podía garantizar si la modificación del plan, seguía haciendo que este fuese óptimo para sus objetivos. El trabajo que se plantea es hacer un modelo reactivo con el AGHMO planteado.

## BIBLIOGRAFIA

- [1] Abascal V, Feijoo P. 2009. Proyecto: “Implementación de Algoritmos Genéticos sobre la plataforma de desarrollo paralelo CUDA”. Facultad de informática. Universidad Complutense de Madrid.
- [2] Alonso S, Cordon O, Fernández de Viana I, Herrera F. La Metaheurística de Optimización Basada en Colonias de Hormigas: Modelos y Nuevos Enfoques. Trabajo realizado en el marco del proyecto “Mejora de Metaheurísticas mediante Hibridación y sus Aplicaciones” de la Universidad de Granada. España.
- [3] Alonso S. 2003. La Metaheurística de Optimización Basada en Colonias de Hormigas: Modelos y Nuevos Enfoques. Proyecto “Mejora de Metaheurísticas mediante Hibridación y sus Aplicaciones” Departamento de Ciencias de la Computación e Inteligencia Artificial, E.T.S. Ingeniería Informática. Universidad de Granada, España.
- [4] Arquero M, Nogueras J, Suz S. 2006. Distribución equilibrada del esfuerzo de cómputo en Algoritmos Genéticos Paralelos. Tesis de grado. Facultad de informática. Universidad Complutense de Madrid.
- [5] Balseiro S. 2007. Logística y Distribución: Algoritmos para problemas de ruteo de vehículos con restricciones de capacidad y ventanas de tiempo. Tesis de ingeniería industrial. Facultad de ingeniería. Universidad de Buenos Aires.
- [6] Barán B, Almirón M. 2001. Colonia de hormigas en un ambiente paralelo asíncrono. Centro Nacional de Computación, Universidad Nacional de Asunción, San Lorenzo, Paraguay.
- [7] Barán B, Hermosilla A. 2003. Comparación de un Sistema de Colonias de Hormigas y una Estrategia Evolutiva para el Problema del Ruteo de Vehículos

- con Ventanas de Tiempo en un Contexto Multiobjetivo. Centro Nacional de Computación, Universidad Nacional de Asunción, San Lorenzo, Paraguay.
- [8] Barán B, Schaerer M. 2003. A multiobjective Ant Colony System for Vehicle Routing Problem with Time Windows. Proceedings of the 21st IASTED International Conference APPLIED INFORMATICS. Innsbruck, Austria.
- [9] Barcos L, Rodríguez V, Alvares M. 2002. Algoritmo basado en la optimización mediante colonias de hormigas para la resolución del problema del transporte de carga desde varios orígenes a varios destinos. Departamento de Organización Industrial, Tecnun, Universidad de Navarra, España. V congreso de ingeniería de transporte © *Santander - CIT 2002*.
- [10] Barril J. 2005. Algoritmos meméticos y su aplicación en fixtures deportivos. Memoria de grado. Pontificia Universidad Católica de Valparaíso. Escuela De Ingeniería Informática.
- [11] Bandyopadhyay, Saha, Maulik, Deb. 2007. A Simulated Annealing Based Multiobjective Optimization Algorithm: AMOSA. Department of Computer Science and Engineering, Jadavpur University, India.
- [12] Cordón O, Herrera F, Moreno L. 1999. Integración de Conceptos de Computación Evolutiva en un Nuevo Modelo de Colonia de Hormigas. Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA'99), 1999: p. 98-104.
- [13] Cruz-Chávez, Marco A, Díaz-Parra, Ocotlán. 2009. Un Mecanismo de Vecindad con Búsqueda Local y Algoritmo Genético para el Problema de Transporte con Ventanas de Tiempo. Programación Matemática y Software (2009), Vol. 1. No 1. ISSN: En trámite.
- [14] Donoso Y. 2005. Multiobjective optimization scheme for static and dynamic multicast flows. PhD Thesis. Department of Electronics, Computer science and Automatic control. Universidad de Girona. España.
- [15] Donoso Y, Albor P, Benavides A. 2005. Optimización con múltiples objetivos utilizando Tabú Search. Investigación "Aplicación de ingeniería de

- tráfico en redes multicast”. Departamento de Ingeniería de Sistemas y Computación, Universidad del Norte.
- [16] Escobar D. 2008. Instrumentos Y Metodología De Planes De Movilidad Y Transporte En Las Ciudades Medias Colombianas. Tesis Doctoral en gestión del territorio e infraestructuras del transporte. Universidad Politécnica De Cataluña. España.
- [17] Fernández J. 2008. Extensión de algoritmo evolutivo MOS con técnicas de evolución diferencial. Tesis de Grado. Facultad Informática. Universidad Politécnica de Madrid. España.
- [18] Galvis J, Garcés A, Escobar A. Diciembre 2005. Aplicación Del Algoritmo De Búsqueda Tabú Al Problema De Despacho Hidrotérmico. Scientia Et Technica, Vol. XI, Núm. 29, pp. 25-30. Universidad Tecnológica de Pereira, Colombia.
- [19] García A. 2007. Programación del transporte de hidrocarburos por oleoductos mediante la combinación de técnicas metaheurísticas y simulación. Tesis de grado. Departamento de ingeniería de organización, administración de empresas y estadística. Escuela técnica superior de ingenieros industriales. Universidad Politécnica de Madrid.
- [20] García F, Melián B, Moreno J.A., 2002. Búsqueda Dispersa para problemas de localización de p servicios con objetivos múltiples. En: II Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirado (pp. 146-152).
- [21] García F, García-Torres M, Moreno J. 2003. Búsqueda Dispersa para el Problema de la Selección de Variables. CAEPIA-2003.
- [22] García R. 2001. Metodología para el diseño de redes de transporte y para la elaboración de algoritmos en programación matemática convexa diferenciable. Tesis Doctoral. Departamento matemáticas aplicadas y estadística. Escuela técnica superior de ingenieros aeronáuticos. Universidad Politécnica de Madrid
- [23] Gómez O, Casado S, Núñez L, Pacheco J. 2004. Resolución del problema de selección de variables cuantitativas mediante GRASP Aplicación a ratios

financieros. Fac. C Económicas y Empresariales. Universidad de Burgos, España.

- [24] Gruttner E, Pinninghoff M. 2003. Algoritmos Genéticos en Recorridos óptimos de líneas de Transporte público. Departamento de Ingeniería informática y Ciencias de la computación. Universidad de Concepción. Chile
- [25] Hermosilla A, Barán B. 2003. Comparación de un sistema de colonias de hormigas y una estrategia evolutiva para un Problema Multiobjetivo de Ruteo de Vehículos con Ventanas de Tiempo. Universidad Nacional de Asunción, Centro Nacional de Computación, San Lorenzo, Paraguay.
- [26] Hernández J, García M, Korstanje A, Contreras S. 2002. El problema de transporte de múltiples productos, incluyendo los transportes. Una aproximación con multiatributo. Documento presentado en el XI Congreso Latino-iberoamericano de Investigación de Operaciones, Concepción, Chile.
- [27] Hertz A., Jaumard B., Ibeiro C, Formosinho Filho, W.P., 1994. A multicriteria tabu search approach to cell formation problems in group technology with multiple objectives. *Recherche Operationelle/Operations Research* 28 (3), 303–328.
- [28] Kulturel-Konak S, Smith A, Norman B. 2006. Multi-objective tabu search using a multinomial probability mass function. *European Journal of Operational Research* 169 (2006) 918-931.
- [29] K. Smith, R. Everson, and J. Fieldsend, .Dominance measures for multi-objective simulated annealing,. in *Proceedings of the 2004 IEEE Congress on Evolutionary Computation (CEC'04)*, 2004, pp. 23.30.
- [30] LaTorre A, Peña J, Fernández J, Muelas S. 2009. MOS como Herramienta para la Hibridación de Algoritmos Evolutivos. Presentado en el marco del VI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'09).
- [31] Martínez J, Almeida F, Giménez D. 2006. Un algoritmo de búsqueda dispersa para el problema de la asignación de procesos en arquitecturas

paralelas heterogéneas. En página web:  
<http://servinf.dif.um.es/~domingo/07/MAEB/articulo.pdf>

- [32] Mauttone A. 2005. Optimización de Recorridos y Frecuencias en Sistemas de Transporte Público Urbano Colectivo. Tesis de Maestría en Informática PEDECIBA. Departamento de Investigación Operativa, Instituto de Computación. Facultad de Ingeniería. Universidad de La República.
- [33] Medina J, Yépes V. 2004. Optimización de rutas mediante la búsqueda de entornos variables y aceptación de umbrales estocásticos. Departamento de ingeniería e infraestructuras de los transportes. Universidad politécnica de Valencia. España.
- [34] Méndez A, Pontin M, Ziletti M, Chávez Luis. 2005. Heurísticas Para La Resolución De Un Problema De Ruteo De Vehículos Periódico Real. Mecánica computacional Vol. XXIV A. Larreteguy (Editor). MECOM 2005 V congreso argentino de mecánica computacional. Buenos Aires, Argentina.
- [35] Mezura E, Coello C. 2002. Conceptos de Optimización Multiobjetivo para el Manejo de Restricciones en Algoritmos Evolutivos: Un Estudio Comparativo. CINVESTAV-IPN. Departamento de Ingeniería Eléctrica. Instituto Politécnico Nacional. México D.F.
- [36] Molina A, Redondo, Ortega M. 2003. Un método semiautomático basado en algoritmos genéticos para el análisis de experiencias de aprendizaje colaborativo. Dpto. de Informática. Universidad de Castilla. España.
- [37] Moscato P, Cotta C. 2003. An Introduction to Memetic Algorithms. School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, Australia.
- [38] Nebro A, Luna F, Alba E, Dorronsoro B, Drillo J. 2005. Un algoritmo multiobjetivo basado en búsqueda dispersa. En página web:  
<http://neo.lcc.uma.es/staff/paco/pdfs/AbYSS.pdf>
- [39] Nebro A, Luna F, Alba E, Beham A. 2006. Un estudio de la aplicación del algoritmo de búsqueda dispersa a optimización multiobjetivo. Institute for Formal Models and Verification. Johannes Kepler University. Linz, Austria

- [40] Obdulia G. Herramientas de cribado virtual aplicadas a inhibidores de tirosina quinasas. Contribución al desarrollo del programa PRALINS para el diseño de quimiotecas combinatorias. Tesis doctoral. Departamento Química Orgánica y Bioquímica. Universitat Ramon Llull Fundació de Catalunya. España.
- [41] Paciello J, Martínez H, Lezcano C, Barán B. 2006. Algoritmos de Optimización multiobjetivos basados en colonias de hormigas. Universidad Nacional de Asunción, Centro Nacional de Computación. San Lorenzo, Paraguay.
- [42] Pilegaard M. 1996. Tabu Search for Multiobjective optimization: MOTS. Institute of Mathematical Modelling. Technical University of Denmark. Presented at *MCDM '97, Cape Town, South Africa, January 6-10, 1997*
- [43] Ramírez N. 2006. Una nueva propuesta para optimización multiobjetivo basada en búsqueda dispersa (Scatter Search). Tesis de maestría. Centro de investigación y de estudios avanzados del instituto politécnico nacional. Departamento de computación
- [44] Redondo M.A., Bravo C., Bravo J, Ortega M. 2003. Applying Fuzzy Logic to Analyze Collaborative Learning Experiences. Journal of the United States. Distance Learning Association. Special issue on Evaluation Techniques and Learning Management Systems. Vol. 17, N° 2, pp, 19-28. USA
- [45] Saavedra I. 2008. Diseño y desarrollo del sistema de soporte a decisiones de planificación de rutas para el transporte terrestre de pasajeros, modelos de programación de flota, conductores, mantenimiento y abastecimiento de combustible. Tesis de Maestría en Gestión Industrial. Departamento de Ingeniería Industrial. Universidad del Norte. Barranquilla, Colombia.
- [46] Sánchez L, Villar J. 2007. Obtaining transparent models of chaotic systems with multiobjective simulated annealing algorithms. Computer Science Department, Universidad de Oviedo. España.
- [47] Sanz J. 2001. Control de procesos basados en eventos mediante bases de conocimiento borrosas y algoritmos genéticos. Tesis Doctoral. Escuela

técnica superior de ingenieros industriales. Universidad Politécnica de Madrid. España.

- [48] Servin O. 2004. Equilibrado de líneas de ensamble en la industria del vestido: un enfoque mediante algoritmos genéticos híbridos. Tesis de Maestría con énfasis en ciencias de ingeniería textil. Escuela superior de ingeniería textil. Instituto Politécnico Nacional. México DF.
- [49] Suman B, 15 April 2005. Study of self-stopping PDMOSA and performance measure in multiobjective optimization,. *Computers and Chemical Engineering*, vol. 29, no. 5, pp. 1131.1147.
- [50] Torres F. Algoritmo genético basado en una heurística de inserción para el problema de ruteo de helicópteros. Universidad de Los Andes, Departamento de Ingeniería Industrial, Bogotá, Colombia.
- [51] Villagra M. 2001. Optimización Multiobjetivo con Técnicas de Inteligencia Artificial. Facultad de Ciencias y Tecnología, Universidad Católica de Asunción. Paraguay.
- [52] Yépes V, Medina J. 2003. Optimización del problema generalizado de las rutas con restricciones temporales y de capacidad (CVRPSTW). Departamento de ingeniería e infraestructuras de los transportes. Universidad politécnica de Valencia. España.