

METODOLOGÍA PARA LA OPTIMIZACIÓN DE MÚLTIPLES OBJETIVOS BASADA EN AG Y USO DE PREFERENCIAS

BIBIANA ANDREA CUARTAS TORRES



**UNIVERSIDAD NACIONAL DE COLOMBIA
FACULTAD DE MINAS
ESCUELA DE SISTEMAS
MEDELLÍN
2009**

METODOLOGÍA PARA LA OPTIMIZACIÓN DE MÚLTIPLES OBJETIVOS BASADA EN AG Y USO DE PREFERENCIAS

BIBIANA ANDREA CUARTAS TORRES

Tesis de grado
presentada como requisito parcial para optar el título de
Magister en Ingeniería - Ingeniería de Sistemas

Directora:
PATRICIA JARAMILLO ALVAREZ, Ph.D



UNIVERSIDAD NACIONAL DE COLOMBIA
FACULTAD DE MINAS
ESCUELA DE SISTEMAS
MEDELLÍN
2009

A Dios, Fuente de toda sabiduría

AGRADECIMIENTOS

A mis padres, Héctor y Liney, por ser una bendición en mi vida, por su esfuerzo y apoyo en todas mis decisiones.

A mi directora, Patricia Jaramillo, por ser de gran apoyo en mi vida profesional y porque sin ella no hubiese sido posible este trabajo.

A mis profesores, Yris Olaya, Santiago Arango, Carlos Jaime Franco, John William Branch, Juan David Velásquez, Fernando Arango, y demás profesores de la Escuela de Sistemas quienes siempre recordaré como grandes personas que aportaron a mi vida profesional.

A mis amigas y compañeras Lisbet y Sofia por su apoyo y amistad incondicional la cual hizo que yo disfrutara de este proceso.

A Juan Esteban, por su valiosa colaboración, mil gracias.

Y a todas aquellas personas que me acompañaron en estos años en mi vida universitaria y que de alguna manera contribuyeron al éxito de éste trabajo.

RESUMEN

En este trabajo se considera la solución de problemas multiobjetivo a través de algoritmos genéticos, el cual consiste en hacer búsquedas aleatorias en el espacio de búsqueda señalado por las restricciones, obteniendo soluciones cada vez más eficientes. Para lograrlo, se proponen dos nuevas metodologías, la primera (AGEM) que considera el elitismo como un concepto interesante para no perder los buenos resultados que se hayan logrado y obtener una frontera de Pareto cercana a la real y la segunda (AGEM-P) que considera las preferencias del decisor de una forma interactiva, tal que el decisor puede dirigir la búsqueda del algoritmo hacia la zona de su interés. AGEM obtienen mejores soluciones en el problema de la mochila comparándola con algoritmos como el SPEA2 y NSGAI y AGEM-P le permite al decisor obtener solo una porción de la frontera de Pareto conforme a sus preferencias adquiriendo conocimiento del problema, tal que para él le será mucho más fácil decidir entre este pequeño grupo de alternativas. Además de ello, para el caso de la mochila, AGEM-P le ofrece al decisor alternativas de solución que no considera un algoritmo sin preferencias y mucho más cercanas a la frontera de Pareto real, ya que al restringir la zona de búsqueda con las preferencias, aprovecha el costo computacional en buscar soluciones más eficientes en vez de buscar en zonas que ya no son de interés para el decisor.

PALABRAS CLAVES: Algoritmos genéticos multiobjetivo, algoritmo genético interactivo, preferencias del decisor, optimización.

ABSTRACT

In this work the solution of multi-objective problems through genetic algorithms is considered, which consists of making random searches in the space of search pointed out by the constraints, getting increasingly efficient solutions. To achieve this, two new methodologies are proposed, the first one (AGEM) considers the elitism as an interesting concept to keep the good results that were achieved and to obtain a Pareto frontier close to the real one and the second (AGEM-P) that considers the preferences of the decision maker in an interactive manner, such that, the decision maker can guide the search of algorithms to the area of his interest. AGEM obtains better solutions in the knapsack problem when it is compared with algorithms such as SPEA2 and NSGAI and AGEM-P allows the decision maker to obtain only a portion of the Pareto frontier according to their preferences and acquire knowledge of the problem such that the decision maker will be much easier to decide between this small group of alternatives. Furthermore, in the case of the knapsack, AGEM-P provides alternatives of solution to the decision maker that an algorithm without preferences doesn't consider and those are much closer to the real Pareto frontier, because by restricting the search area with the preferences, take advantage of the computational cost to search more efficient solutions rather than look for areas that are no longer relevant for the decision-maker.

KEY WORDS: Multi-objective genetic algorithms, interactive genetic algorithms, preference by decision maker, optimization.

TABLA DE CONTENIDO

1	INTRODUCCIÓN	13
1.1	PROBLEMÁTICA	13
1.2	ANTECEDENTES DEL PROBLEMA.....	14
1.3	OBJETIVOS	15
1.3.1	<i>Objetivo General</i>	15
1.3.2	<i>Objetivos Específicos</i>	15
1.4	CONTENIDO	16
2	GENERALIDADES DE PROBLEMAS MULTIOBJETIVO	17
2.1	INTRODUCCIÓN	17
2.2	PROBLEMAS DE OPTIMIZACIÓN MULTIOBJETIVO	17
2.3	TÉCNICAS PARA RESOLVER PROBLEMAS MULTIOBJETIVO	20
2.3.1	<i>Técnicas A-Priori</i>	20
2.3.1.1	Programación por compromiso.....	20
2.3.1.2	Programación por metas.....	21
2.3.1.3	Funciones de utilidad o de valor multiatributo.....	22
2.3.2	<i>Técnicas Interactivas</i>	23
2.3.2.1	Método Stem.....	24
2.3.2.2	Método Visual Interactive Approach – VIA	25
2.3.2.3	Método Ecuador.....	26
2.3.3	<i>Técnicas A-Posteriori</i>	27
2.3.3.1	Métodos de ponderación.....	28
2.3.3.2	Método de las restricciones.....	28
2.4	PROBLEMAS MULTIOBJETIVO DIFÍCILES DE RESOLVER	29
2.4.1	<i>Problemas no Lineales</i>	29
2.4.2	<i>Problemas combinatoriales</i>	30
2.5	TÉCNICAS META-HEURÍSTICAS	30
2.6	ALGORITMOS EVOLUTIVOS O COMPUTACIÓN EVOLUTIVA	31
2.6.1	<i>Programación Evolutiva</i>	31
2.6.2	<i>Estrategia Evolutiva</i>	32
2.6.3	<i>Algoritmos Genéticos</i>	32
2.7	CONCLUSIONES.....	33
3	ALGORITMOS GENÉTICOS	35

3.1	INTRODUCCIÓN	35
3.2	DESCRIPCIÓN DE UN ALGORITMO GENÉTICO.....	35
3.3	SELECCIÓN	38
3.3.1	<i>Selección Proporcional</i>	38
3.3.1.1	Ruleta.....	38
3.3.1.2	Sobrante Estocástico	39
3.3.1.3	Universal Estocástica	40
3.3.1.4	Muestreo Determinístico	41
3.3.1.5	Selección por Jerarquías.....	42
3.3.2	<i>Selección mediante Torneo</i>	43
3.3.2.1	Determinístico	43
3.3.2.2	Probabilístico	44
3.3.3	<i>Selección de Estado Uniforme</i>	44
3.4	CRUZAMIENTO.....	44
3.4.1	<i>Cruza de un punto</i>	45
3.4.2	<i>Cruza de dos puntos</i>	45
3.4.3	<i>Cruza uniforme</i>	45
3.5	MUTACIÓN.....	46
3.6	CONCLUSIONES.....	47
4	ALGORITMOS GENÉTICOS PARA RESOLVER PROBLEMAS DE OPTIMIZACIÓN	
	MULTIOBJETIVO	49
4.1	INTRODUCCIÓN	49
4.2	ALGORITMOS GENÉTICOS MULTIOBJETIVO	49
4.2.1	<i>VEGA</i>	50
4.2.2	<i>MOGA</i>	50
4.2.3	<i>NPGA</i>	51
4.2.4	<i>NSGA</i>	52
4.2.5	<i>SPEA</i>	53
4.2.6	<i>NSGA-II</i>	55
4.2.7	<i>SPEA2</i>	56
4.2.8	<i>cNSGA-II</i>	58
4.3	OBSERVACIONES SOBRE LOS MÉTODOS.....	58
4.4	INCORPORACIÓN DE LAS PREFERENCIAS EN LOS ALGORITMOS GENÉTICOS	
	MULTIOBJETIVO.....	59
4.4.1	<i>Algoritmos A priori</i>	60
4.4.2	<i>Algoritmos Interactivos</i>	62
4.4.3	<i>Algoritmos A posteriori</i>	63
4.5	DEBILIDADES Y FORTALEZAS ENCONTRADAS EN LAS METODOLOGÍAS CON	
	PREFERENCIAS	64
4.5.1	<i>Falencias encontradas en la literatura</i>	65
4.5.2	<i>Fortalezas encontradas en la literatura</i>	66

4.6	CARACTERÍSTICAS PARA UNA NUEVA METODOLOGÍA.....	67
4.7	CONCLUSIONES.....	67
5	METODOLOGÍA PROPUESTA.....	69
5.1	INTRODUCCIÓN	69
5.2	ALGORITMO GENÉTICO ELITISTA MULTIOBJETIVO – AGEM	69
5.2.1	<i>Manejo de las restricciones</i>	70
5.2.2	<i>Dominancia</i>	70
5.2.3	<i>Factor de Distancia</i>	70
5.2.4	<i>Medida de desempeño – Aptitud</i>	71
5.2.5	<i>Archivo externo ELITE</i>	71
5.2.6	<i>Selección de Padres</i>	72
5.2.7	<i>Operadores de Cruzamiento y Mutación</i>	72
5.2.8	<i>Seleccionar la nueva generación</i>	72
5.2.9	<i>Diagrama</i>	72
5.3	ALGORITMO GENÉTICO ELITISTA MULTIOBJETIVO QUE INCORPORA LAS PREFERENCIAS DEL DECISOR – AGEM-P	75
5.3.1	<i>Factor de Preferencia</i>	79
5.3.2	<i>Dominancia por preferencia</i>	80
5.3.3	<i>Función de Evaluación – Aptitud</i>	81
5.3.4	<i>Archivo Externo ELITE</i>	82
5.3.5	<i>Seleccionar la nueva generación</i>	83
5.4	CONCLUSIONES.....	83
6	RESULTADOS DE LA SIMULACIÓN	85
6.1	INTRODUCCIÓN	85
6.2	PROBLEMA TEST	85
6.3	MÉTRICAS DE EVALUACIÓN	87
6.3.1	<i>Distancia Generacional (Generational Distance – GD)</i>	87
6.3.2	<i>Distribución (Spacing – S)</i>	87
6.4	CONFIGURACIÓN DEL ALGORITMO	88
6.5	RESULTADOS.....	89
6.5.1	<i>Resultados experimentales de AGEM</i>	89
6.5.2	<i>Resultados experimentales de AGEM-P</i>	93
6.6	CONCLUSIONES.....	99
7	CONCLUSIONES Y TRABAJOS FUTUROS.....	101
8	BIBLIOGRAFÍA.....	105
	ANEXOS	108

LISTA DE GRÁFICAS

<i>Gráfica 1. Representación de la Frontera de Pareto</i>	<i>19</i>
<i>Gráfica 2. Sharing</i>	<i>52</i>
<i>Gráfica 3. Limitando el espacio de búsqueda con los mínimos esperados.....</i>	<i>78</i>
<i>Gráfica 4. Función del Factor de Preferencia.....</i>	<i>79</i>
<i>Gráfica 5. Dominancia teniendo en cuenta las preferencias del decisor</i>	<i>81</i>
<i>Gráfica 6. Métrica GD para 2 objetivos 400 variables</i>	<i>90</i>
<i>Gráfica 7. Métrica S para 2 objetivos 400 variables.....</i>	<i>90</i>
<i>Gráfica 8. Métrica GD para 2 objetivos 600 variables</i>	<i>91</i>
<i>Gráfica 9. Métrica S para 2 objetivos 600 variables.....</i>	<i>91</i>
<i>Gráfica 10. Métrica GD para 2 objetivos 800 variables</i>	<i>91</i>
<i>Gráfica 11. Métrica S para 2 objetivos 800 variables.....</i>	<i>91</i>
<i>Gráfica 12. Métrica GD para 3 objetivos 400 variables</i>	<i>92</i>
<i>Gráfica 13. Métrica S para 3 objetivos 400 variables.....</i>	<i>92</i>
<i>Gráfica 14. Métrica GD para 3 objetivos 600 variables</i>	<i>92</i>
<i>Gráfica 15. Métrica S para 3 objetivos 600 variables.....</i>	<i>92</i>
<i>Gráfica 16. Métrica GD para 3 objetivos 800 variables</i>	<i>93</i>
<i>Gráfica 17. Métrica S para 3 objetivos 800 variables.....</i>	<i>93</i>
<i>Gráfica 18. Información que observa el decisor antes de definir sus preferencias.....</i>	<i>94</i>
<i>Gráfica 19. Rango del Objetivo 1.....</i>	<i>94</i>
<i>Gráfica 20. Algoritmo pausado para redefinir preferencias.....</i>	<i>95</i>
<i>Gráfica 21. Movimiento de la Elite hacia la nueva zona de búsqueda.....</i>	<i>95</i>
<i>Gráfica 22. Proceso de búsqueda de NSGAll para el problema de la Mochila con 400, 600 y 800 variables y 2 Objetivos</i>	<i>96</i>
<i>Gráfica 23. Proceso de búsqueda de SPEA2 para el problema de la Mochila con 400, 600 y 800 variables y 2 Objetivos</i>	<i>97</i>
<i>Gráfica 24. Proceso de búsqueda de AGEM para el problema de la Mochila con 400, 600 y 800 variables y 2 Objetivos</i>	<i>98</i>
<i>Gráfica 25. Comparación entre AGEM y AGEM-P para extremos en el problema de la Mochila con 2 Objetivos y 400 Variables.....</i>	<i>98</i>
<i>Gráfica 26. Comparación entre AGEM y AGEM-P para zonas en común en el problema de la mochila con 2 Objetivos y 400 Variables</i>	<i>99</i>

LISTA DE DIAGRAMAS

<i>Diagrama 1. Representación de una solución en los algoritmos genéticos</i>	<i>35</i>
<i>Diagrama 2. Representación del proceso de un Algoritmo Genético.....</i>	<i>37</i>
<i>Diagrama 3. Representación de Cruzamiento en un punto</i>	<i>45</i>
<i>Diagrama 4. Representación de Cruzamiento en dos puntos</i>	<i>46</i>
<i>Diagrama 5. Representación de Cruzamiento Uniforme.....</i>	<i>46</i>
<i>Diagrama 6. Representación de Mutación.....</i>	<i>47</i>
<i>Diagrama 7. Primera iteración de AGEM</i>	<i>73</i>
<i>Diagrama 8. Representación en diagrama de flujo del algoritmo AGEM.....</i>	<i>74</i>

LISTA DE TABLAS

Tabla 1. Parámetros de los Algoritmos 88

LISTA DE ANEXOS

<i>ANEXO A. Manual de Usuario: Herramienta basada en AGEM-P para resolver el problema de la Mochila.....</i>	<i>108</i>
<i>ANEXO B. Herramienta Interactiva para resolver el problema de la mochila con el método AGEM – P.....</i>	<i>111</i>
<i>ANEXO C. Datos para el problema de la Mochila.....</i>	<i>112</i>

1 INTRODUCCIÓN

1.1 PROBLEMÁTICA

Hoy en día la toma de decisiones es un proceso que ha cobrado importancia dado que el mundo en el que estamos inmersos cambia rápidamente obligando a la(s) persona(s) a tomar decisiones (simples o complejas) en cortos periodos de tiempo.

Tomar una decisión es un proceso que se hace complejo ya que muchas veces quien toma la decisión no cuenta con un solo interés sino con múltiples intereses u objetivos que desea cumplir y por lo general estos están en conflicto entre sí. Más aún, si a esto se le suma que los sistemas donde toman las decisiones son sistemas dinámicos, con información imprecisa, en ambiente de incertidumbre y en algunos casos, las decisiones son tomadas por un grupo de personas con intereses diferentes.

La investigación de operaciones es una ciencia que modela este tipo de problemas haciendo uso de la matemática y la lógica. Intenta encontrar una mejor solución (solución óptima) o varias mejores soluciones, tal que conduzca las operaciones dentro de una organización asignando de la manera más eficaz los recursos disponibles en diferentes actividades (Hillier y Lieberman, 2002).

Esta ciencia proporciona diferentes herramientas a las personas que van a tomar una decisión, las cuales dependen de las características de los problemas que se desean resolver. Entre estas herramientas están la programación lineal, programación dinámica, programación entera, programación no lineal, teoría de juegos, análisis de decisiones, procesos estocásticos, simulación, entre otras.

Particularmente, el análisis de decisiones estudia la toma de decisiones con un grado de incertidumbre. Ésta proporciona al decisor metodologías que indican como tomar la mejor decisión de forma racional cuando se presenta incertidumbre en la información y por tanto en los resultados (Hillier y Lieberman, 2002).

Algunas de estas metodologías son el análisis multiobjetivo, análisis multiobjetivo con múltiples decisores, decisiones bajo certidumbre, decisiones bajo riesgo, decisiones bajo incertidumbre, entre otras (Hillier y Lieberman, 2002).

El interés de esta investigación se enmarca dentro del análisis multiobjetivo con un solo decisor, el cual es considerado un tópico de investigación muy importante, no solo porque este tipo de problema es muy común en la vida real, sino porque todavía hay muchas preguntas abiertas en esta área (Abbass et al., 2001).

1.2 ANTECEDENTES DEL PROBLEMA

En el estado del arte del análisis multiobjetivo se pueden encontrar diversas metodologías que describen algoritmos clásicos de optimización y que son muy eficientes y efectivos a la hora de resolver este tipo de problemas. Sin embargo, en el mundo real existen problemas que las metodologías clásicas no pueden resolver con facilidad, ya sea porque tienen espacios de búsqueda sumamente grandes que el algoritmo se vuelve ineficiente, porque el planteamiento haga uso de ecuaciones no lineales difícil de resolver o porque simplemente no son fáciles de plantear matemáticamente y por tanto no se puede aplicar una de estas metodologías.

La existencia de ese tipo de problemas ha llevado a explorar técnicas meta-heurísticas que ofrezcan al decisor soluciones óptimas o por lo menos cercanas a ellas en el menor tiempo posible. Entre las más reconocidas están los Algoritmos Evolutivos, Búsqueda Tabú, Recocido Simulado, Colonia de Hormigas, Nube de Partículas, entre otras, las cuales representan modelos provenientes de la física, biología o la genética.

En la práctica, una de las técnicas más utilizadas para resolver problemas multiobjetivo han sido los algoritmos genéticos, categorizados en la familia de los algoritmos evolutivos. Su alta aplicabilidad se debe a su búsqueda poblacional eficiente, ya que trabajan con varios individuos de manera simultánea y hallan varias soluciones cada vez más eficientes y satisfactorias en una sola corrida.

Desde 1967 con Rosenberg, se mencionaron los algoritmos genéticos para resolver problemas multiobjetivo. Más tarde, en 1983, el investigador David Schaffer plantea un algoritmo llamado VEGA (Schaffer, 1985) que luego fue criticado por David Goldberg en 1989 quien plantea la necesidad de hacer uso del concepto de dominancia para mover la población hacia la frontera de Pareto. En la década de los 90's surgieron muchas metodologías que buscaban acercarse cada vez más a la frontera real de una manera eficiente, tales como MOGA (Fonseca y Fleming, 1993), NPGA (Horn et al., 1994), NSGA (Srinivas y Deb, 1994), SPEA (Zitzler y Thiele, 1999), entre otras. Pero en el 2000 y el 2001 NSGAI (Deb et al., 2002) y SPEA2 (Zitzler et al., 2001) aumentaron en gran manera la efectividad de este tipo de algoritmos, y a partir de allí se han propuesto diferentes metodologías, aunque no han sido tan reconocidas por la comunidad científica como estas dos últimas.

Sin embargo, con este tipo de metodologías, el problema de toma de decisiones no se resuelve totalmente, ya que al final éstas ofrecen al decisor un número bastante grande de alternativas que pueden ser satisfactorias para él, pero no le ofrecen una única solución; por tanto el decisor debe continuar resolviendo un problema de decisión discreto teniendo en cuenta sus preferencias.

En el estado del arte (Coello, 2000), (Rachmawati y Srinivasan, 2006) se ha considerado el uso de las preferencias no sólo al finalizar la búsqueda del algoritmo, sino antes o durante la búsqueda, con el fin de lograr soluciones satisfactorias para el decisor conforme a sus preferencias. Sin embargo, se debe tener en cuenta que las preferencias de un decisor son relativas y cambiantes, puesto que son sujetas al ambiente que percibe en un momento determinado, por tanto es riesgoso considerar que el decisor pueda decidir acerca de sus preferencias cuando no tiene un conocimiento previo del problema.

Esta tesis pretende realizar un aporte teórico tanto en las metodologías de algoritmos genéticos multiobjetivo como en el uso de preferencias de manera interactiva, ya que se considera un tópico muy importante a desarrollar.

1.3 OBJETIVOS

1.3.1 Objetivo General

Hacer uso de las fortalezas de las técnicas interactivas para proponer una metodología multiobjetivo basada en algoritmos genéticos que le permita al decisor intervenir de manera efectiva en el algoritmo, tal que éste, al final le proporcione un número pequeño de alternativas de solución cercanas a la óptima y en la zona de su interés.

1.3.2 Objetivos Específicos

- Proponer una nueva metodología para resolver problemas multiobjetivo basada en algoritmos genéticos sin preferencias que mejore los resultados obtenidos por las técnicas ya existentes.
- Determinar el momento en que el decisor debe intervenir en el algoritmo con preferencia, de tal forma que lo haga de una manera eficiente y práctica luego de estar adquiriendo conocimiento del problema.
- Identificar si es necesario que el decisor modifique parámetros del algoritmo genético mientras define sus preferencias.
- Ofrecer información al decisor, tal que le permita ir adquiriendo conocimiento del problema.

Introducción

- Validar las metodologías propuestas en un problema test evaluando su eficiencia mediante métricas seleccionadas previamente y a través de gráficos.

1.4 CONTENIDO

A continuación, en el capítulo 2 se describe el problema de optimización multiobjetivo, mencionando algunas técnicas clásicas de optimización al igual que algunas técnicas meta – heurísticas que son utilizadas para resolver este tipo de problemas. Dado que el interés de esta investigación son los algoritmos genéticos, se profundizará en este tipo de metodología en el capítulo 3. En el capítulo 4 se describirán las principales metodologías de algoritmos genéticos multiobjetivo y se mencionará cómo se han involucrado las preferencias del decisor. Teniendo en cuenta las debilidades y fortalezas de estas metodologías, en el capítulo 5 se describe una propuesta metodológica interactiva donde el decisor interviene en el proceso de búsqueda para obtener mejores resultados y acuerdos a sus preferencias; no sin antes proponer otra nueva metodología de algoritmos genéticos multiobjetivo que obtenga mejores resultados comparándola con las ya existentes. Luego, en el capítulo 6 se presentarán los resultados obtenidos con las dos propuestas metodológicas. Y finalmente en el capítulo 7, se presentan las conclusiones y algunas ideas de posibles trabajos futuros.

2 GENERALIDADES DE PROBLEMAS MULTIOBJETIVO

2.1 INTRODUCCIÓN

En los procesos de toma de decisiones en el mundo real bajo el concepto de optimización, ha sido necesario tener en cuenta varios objetivos de manera simultánea. En este tipo de problemas no se cuenta con una única solución, sino con un conjunto de soluciones que son óptimas o cercanas a serlo.

Dado que finalmente se debe escoger una única solución, el decisor debe hacer uso de sus preferencias con respecto a los criterios que se están evaluando para seleccionar la solución que más se ajusta a sus intereses. Dependiendo la forma como esté interesado en involucrar sus preferencias, si antes, durante o después del proceso de optimización, el decisor podrá resolver su problema multiobjetivo con una técnica específica.

Dentro de la optimización multiobjetivo existen problemas donde a las técnicas clásicas le es difícil encontrar una buena solución, este tipo de problemas han sido tratados con técnicas meta-heurísticas hallando soluciones satisfactorias para el decisor.

En la siguiente sección (Sección 2.2) se describirá la definición matemática de un problema de optimización multiobjetivo, luego en la sección 2.3 se mencionarán algunas de las técnicas utilizadas para resolver este tipo de problema teniendo en cuenta el momento en que el decisor involucre sus preferencias. En la sección 2.4 se mencionaran dos tipos de problemas difíciles de resolver con técnicas clásicas, y que han sido tratados con técnicas meta-heurísticas, las cuales se mencionan algunas en la sección 2.5. Finalmente en la sección 2.6 se define una técnica meta-heurística específica llamada Algoritmos Evolutivos, a la que pertenecen los algoritmos genéticos los cuales son el objeto de esta investigación.

2.2 PROBLEMAS DE OPTIMIZACIÓN MULTIOBJETIVO

Desde hace muchos años se ha considerado la necesidad de incluir en el proceso de toma de decisiones bajo el esquema de optimización, varios objetivos de manera simultánea, que posiblemente no tienen las mismas unidades entre sí. Esta necesidad ha llevado a la exigencia de hacer uso de herramientas que permitan incluir simultáneamente múltiples

objetivos de manera explícita que pueden ser representados matemáticamente o mediante expresiones cualitativas (Smith et al., 2000).

Los problemas donde se consideran más de un objetivo se les denomina Problemas Multiobjetivo, y su propósito es optimizar todos los objetivos de forma simultánea, los cuales, la mayoría de las veces, se encuentran en conflictos. Cuando se habla de optimalidad, se refiere a encontrar la mejor solución posible o una buena aproximación de ésta, dado un conjunto de limitaciones o restricciones.

Un problema multiobjetivo es representado matemáticamente de la siguiente manera (Abbass et al., 2001):

$$\begin{aligned} & \text{Optimizar } F(\vec{x}) \\ & \text{Sujeto a: } \Omega = \{ \vec{x} \in R^n \mid G(\vec{x}) \leq 0 \} \end{aligned}$$

Donde \vec{x} es un vector de variables de decisión (x_1, \dots, x_n) y $F(\vec{x})$ es un vector de funciones objetivo $(f_1(\vec{x}), \dots, f_m(\vec{x}))$. Las funciones $(f_1(\vec{x}), \dots, f_m(\vec{x}))$ pertenecen a R^n y Ω es un conjunto no vacío perteneciente a R^n . El vector $G(\vec{x})$ representa las restricciones que deben cumplir la solución, estas pueden ser mayores o menores que cero (Abbass et al., 2001).

El propósito de los problemas multiobjetivo es hallar un vector \vec{x}^* que pertenezca a la región Ω y que optimice $F(\vec{x})$, donde cada una de las funciones pueden ser maximizadas o minimizadas, dependiendo el problema.

Un problema multiobjetivo no tiene una única solución eficiente, más bien, tiene un conjunto de soluciones eficientes que no pueden ser consideradas diferentes entre sí. A este conjunto de soluciones se le denomina Frontera de Pareto.

Una solución es eficiente cuando no es dominada, es decir, cuando es al menos tan buena como las otras en todos sus objetivos y es mejor en al menos uno de ellos (Abbass et al., 2001).

El conjunto de soluciones pertenecientes a la frontera de Pareto S , en un problema donde todos sus objetivos necesitan ser maximizados, se puede representar de la siguiente manera (Smith et al., 2000):

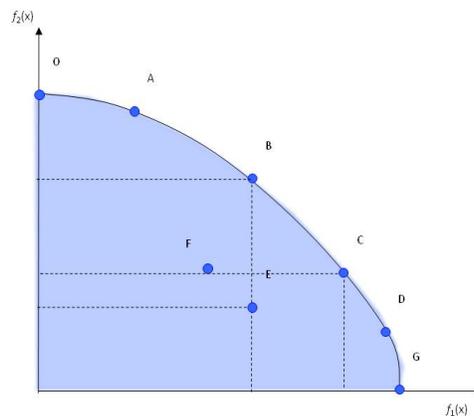
$$\begin{aligned} S = \{ \vec{x} : \vec{x} \in \vec{X}, \text{ no existe otro } \vec{x}^* \in \vec{X} \text{ tal que:} \\ f_i(\vec{x}) > f_i(\vec{x}^*) \text{ para algún } i \in \{1, 2, \dots, m\} \} \end{aligned}$$

$$y f_k(\vec{x}^*) \geq f_k(\vec{x}) \text{ para todo } k \neq i \}$$

Donde m representa el número de objetivos considerados, $f_i(\vec{x})$ es el valor del objetivo i para la solución \vec{x} , y \vec{X} el espacio de soluciones posibles.

Por tanto, si se deseara cambiar de una solución no dominada a otra, dentro del espacio de los objetivos, con el fin de mejorar el valor de uno de ellos, es necesario que uno o más objetivos deban empeorar (Smith et al., 2000).

En el caso de un problema donde se quiere maximizar dos objetivos ($f_1(x)$ y $f_2(x)$), los cuales se pueden dibujar en un plano cartesiano, la frontera de Pareto se puede representar tal como se ilustra en la Gráfica 1 (Smith et al., 2000).



Gráfica 1. Representación de la Frontera de Pareto

Tomado de Smith *et al.* (2000)

El área sombreada es el espacio de soluciones posibles y las soluciones no dominadas están representadas en la curva OABCDG, denominada curva de transformación. En este caso, el punto B tiene mayor valor en el objetivo 2 con relación al punto C, pero tiene menor valor en cuanto al objetivo 1, por tanto no es posible decir que un punto es mejor que el otro. Si se observa, por ejemplo, los puntos F y E que no están dentro de la curva, se puede notar que siempre están dominados por uno o más puntos, por ejemplo el punto E tiene igual valor en el objetivo 1 con relación al punto B, pero tiene menor valor del objetivo 2, por lo cual se dice que E es dominado por B.

Debido a que en un problema real usualmente solo se necesita implementar una solución, en un proceso de toma de decisiones necesariamente debe intervenir un decisor o un grupo de ellos que escoja una entre un conjunto de alternativas eficientes posibles. Para esta investigación solo se tendrá en cuenta el caso donde interviene una sola persona como decisor. Esta persona siempre cuenta con una estructura de preferencias donde representa qué tan importante es para él un objetivo con relación al resto. Esta estructura

siempre influye a la hora de tomar una decisión, ya que el decisor finalmente selecciona la alternativa, entre todas las posibles, que produzca las consecuencias con mayor preferencia para él.

La estructura de preferencias del decisor implica un grado de dificultad, puesto que una decisión afecta simultáneamente diversos atributos que posiblemente compiten entre sí, lo que lleva a que si la decisión tomada pueda satisfacer plenamente la preferencia del decisor en un atributo dado, muy seguramente no lo estará cumpliendo para otro atributo (Smith et al., 2000). Esto hace que para el decisor sea importante que la técnica utilizada para resolver su problema de decisión contenga de una u otra forma la representación de sus preferencias para sentirse a gusto con su resultado.

Para mencionar algunas de las técnicas utilizadas para resolver problemas multiobjetivo, se hará uso de la clasificación propuesta por Cohon y Marks en 1975 (Cohon y Marks, 1975), los cuales las clasifican en técnicas A priori, Interactivas o A posteriori. En el primer grupo las preferencias son involucradas al inicio del proceso; en el segundo, son involucradas en la medida que el algoritmo avanza; y en el último se involucran cuando ya se tiene un grupo de soluciones.

2.3 TÉCNICAS PARA RESOLVER PROBLEMAS MULTIOBJETIVO

2.3.1 Técnicas A-Priori

Las técnicas clasificadas bajo éste enfoque requieren que el decisor represente sus preferencias antes de iniciar el proceso de búsqueda de la solución. Estas pueden estar representadas mediante una solución deseada, una meta propuesta o por ejemplo una función que represente su estructura de preferencia.

El hecho de tener que definir las preferencias antes de iniciar el proceso, hace que para el decisor sea un poco difícil ya que desconoce el comportamiento del problema. Por ejemplo, en el caso de usar una técnica donde el decisor deba definir sus metas a cumplir, éste podría definir las pocas optimista y conseguir unos resultados menos satisfactorios que si definiera una meta más ambiciosa.

Algunas de las técnicas que se clasifican dentro de este grupo son:

2.3.1.1 Programación por compromiso.

Éste método fue propuesto inicialmente por Zeleny (1973), quien propuso hallar una solución o conjunto de ellas que estén más cercana a un punto ideal propuesto por el decisor. Este punto representa lo que el decisor desea obtener en cada objetivo y por lo general, debido al conflicto entre los objetivos, es un punto inalcanzable.

El conjunto de soluciones más cercanas a este punto ideal, llamado conjunto compromiso, se hallan mediante un indicador de distancia, que en el caso de dos objetivos se representa por medio de la siguiente ecuación:

$$L_1 \rightarrow \text{Min } L_1 = \left[\sum_{i=1}^m w_i \frac{f_i^* - f_i(x)}{f_i^* - f_{*i}} \right]$$

Y cuando se tienen más de dos objetivos y se quiere tener una solución que represente mayor equidad entre ellos, en la formula anterior aparece un nuevo parámetro infinito, siendo ahora el indicador de distancia el siguiente:

$$L_\infty \rightarrow \text{Min } L_\infty = d$$

Sujeto a:

$$w_i \frac{f_i^* - f_i(x)}{f_i^* - f_{*i}} \leq d \quad 1 \leq i \leq m$$

Donde w_i representa los factores de ponderación de los objetivos, f_i^* el punto ideal y f_{*i} el punto anti-ideal.

Existen otras métricas de distancia, entre la más común está (Smith et al., 2000):

$$L_k = \left\{ \sum_{i=1}^m w_i^k [f_{i,Max} - f_i(x)]^k \right\}^{1/k}$$

Donde w_i representa la importancia relativa del objetivo i y k el grado del medidor ($1 < k < \infty$) que refleja la importancia de la máxima desviación. Estos dos valores deben ser definidos por el decisor y por tanto representan sus preferencias.

Si los objetivos no están expresados en la misma unidad, entonces estos deben ser re-escalados y para ello la métrica L_k se puede transformar en (Smith et al., 2000):

$$\text{Min} \left\{ L_k(x) = \sum_{i=1}^m w_i^k \left\{ \frac{[f_{i,Max} - f_i(x)]^k}{[f_{i,Max} - f_{i,Min}]^k} \right\} \right\}$$

Esta metodología se puede consultar en Zeleny (1973) o en Smith *et al* (2000).

2.3.1.2 Programación por metas.

Ha sido uno de los métodos más divulgados en el análisis multiobjetivo. Fue desarrollado por Charnes, Cooper y Ferguson (1955) y tiene como propósito minimizar el nivel de

desviación entre la meta lograda del objetivo y una meta inicialmente propuesta por el decisor.

La formulación de programación por metas se expresa como sigue (Smith et al., 2000):

$$\text{Min} \left\{ \sum_{i=1}^m |f_{i,Meta} - f_i(x)| \right\}$$

Sujeto a

$$x \in X$$

Con el fin de convertir la anterior ecuación en una función lineal se crean dos nuevas variables por cada objetivo i d_i^+ y d_i^- las cuales corresponden a la desviación positiva (exceso) y a la desviación negativa (déficit) de $f_i(x)$ con respecto a $f_{i,Meta}$ respectivamente. Con estas nuevas variables, el problema se convierte en:

$$\text{Min} \left\{ \sum_{i=1}^m (d_i^+ + d_i^-) \right\}$$

Sujeto a:

$$f_i(x) + d_i^+ - d_i^- = f_{i,Meta} \quad \forall i = 1, 2, \dots, m$$

$$x \in X$$

Para una mayor profundización de esta metodología, se puede consultar Charnes *et al.* (1955) o Smith *et al.* (2000).

2.3.1.3 Funciones de utilidad o de valor multiatributo

Una función de utilidad o de valor individual es una función que convierte el valor de un objetivo o atributo en otro que representa la preferencia del individuo con relación a ese valor.

La función de valor se diferencia de las de utilidad, porque las primeras se definen en un contexto determinístico y las segundas en uno probabilístico. En el caso de las funciones de valor, el decisor la definirá teniendo en cuenta la importancia que le da a la función objetivo que esté evaluando, en cambio que la función de utilidad la construye determinando las preferencias del decisor ante varios escenarios.

Halladas las funciones de utilidad o de valor para cada objetivo, se puede hallar la función de utilidad o de valor multiatributo. Ésta se consigue a partir de una agregación de las

funciones individuales de cada objetivo. Si se consideran una independencia entre las funciones individuales, esta agregación se puede definir a partir de las siguientes reglas:

Regla aditiva:

$$U(G) = \sum_{i=1}^m k_i u_i(G_i)$$

Regla multiplicativa:

$$1 + kU(G) = \prod_{i=1}^m [1 + k k_i u_i(G_i)]$$

Donde $U(G)$ es la función de utilidad o de valor multiatributo que toma valores entre cero y uno, los k_i , $i = 1, 2, \dots, m$ son constantes positivas y menores que 1 dadas por el decisor y que representan las preferencias de éste, k es una constante que satisface la siguiente ecuación:

$$1 + k = \prod_{i=1}^m [1 + k k_i]$$

Para una mayor profundización en ésta metodología se puede consultar Smith *et al.* (2000).

2.3.2 Técnicas Interactivas

Las técnicas que se clasifican en este grupo son aquellas donde las preferencias del decisor se expresan de forma interactiva durante la corrida del algoritmo, es decir, le permiten al decisor reestructurar sus preferencias si lo desea, luego de conocer las soluciones parcialmente no dominadas.

Dependiendo de la metodología, el decisor proporciona un tipo de información en cada iteración, por ejemplo, puede proporcionar los *tradeoffs* o tasas de intercambio entre los objetivos luego de conocer los valores objetivos en cada iteración; puede seleccionar una alternativa a partir de conocer un conjunto pequeño eficiente de ellas y el valor de sus objetivos o puede proporcionar nuevos valores de referencia de acuerdo a sus preferencias después de conocer los valores alcanzados en cada función objetivo (Caballero et al., 2002).

Cada uno de estos métodos tiene la ventaja de que el decisor tiene la posibilidad de calibrar sus preferencias durante el proceso, ya que muchas veces éstas no se tienen claramente definidas. Sin embargo, dependiendo de la forma como se le solicita

información al decisor en cada iteración, el método puede exigirle un gran esfuerzo mental al decisor.

Algunas de las técnicas que se clasifican dentro de este grupo son las siguientes:

2.3.2.1 Método Stem

Este método fue publicado por los autores R. Benayoun, J. De Montgolfier, J. Tergny y O. Laritchev (1971) y consiste en restringir el conjunto de oportunidades en cada iteración, en función de las preferencias manifestadas por el decisor. Las preferencias se representan relajando un objetivo satisfactorio, con el fin de mejorar otro objetivo que aún no lo es.

Una solución se obtiene luego de minimizar la distancia de *Tchebychev* respecto al vector ideal sobre el conjunto de oportunidades restringido.

El algoritmo consiste en lo siguiente (Caballero et al., 2002):

- Se obtienen los ideales f_i^* $i = 1, \dots, m$ y anti-ideales f_{i^*} $i = 1, \dots, m$.
- Se obtienen los valores de los parámetros π_i (pesos normalizadores):

$$\pi_i = \begin{cases} \frac{f_i^* - f_{i^*}}{f_i^*} \left(\sum_{j=1}^n c_{ij}^2 \right)^{-1/2} & \text{si } f_i^* < 0 \\ \frac{f_i^* - f_{i^*}}{f_{i^*}} \left(\sum_{j=1}^n c_{ij}^2 \right)^{-1/2} & \text{si } f_i^* \geq 0 \end{cases} \quad i = 1, 2, \dots, m$$

Donde c_{ij} son los coeficientes de la función objetivo f_j y se denota por $J^h \equiv$ conjunto índices de las funciones a relajar en la iteración h e inicializa $X^0 = X$, $J^0 = \emptyset$ y $h = 0$.

- Se calculan los pesos de la distancia de la métrica de *Tchebychev*:

$$\mu_i^h = \begin{cases} 0 & \text{si } i \in J^h \\ \frac{\pi_i}{\sum_{j \in J^*} \pi_j} & \text{si } i \in \{1, \dots, m\} - J^h \end{cases}$$

- Se obtiene la solución que minimiza la distancia desde el ideal al espacio objetivo restringido $Z^h = f(X^h)$. Para ello, se resuelve el problema:

$$\begin{aligned} & \min_{\mathbf{x}, \alpha} \alpha \\ \text{s.a. : } & \mu_i (f_i(\mathbf{x}) - f_i^*) \leq \alpha \quad i = 1, \dots, m \\ & \mathbf{x} \in X^h \quad \alpha \geq 0 \end{aligned}$$

Dicha solución se denota por x^h y su correspondiente vector criterio $f^h = f(x^h)$.

- El decisor manifiesta las preferencias sobre la solución:

Si está satisfecho, se termina con (x^h, f^h) como solución final. En caso contrario, se continúa.

- Si se continua se toma $h = h + 1$. El decisor especifica:
 - Funciones a mejorar (f_i con $i \in \{1, \dots, m\} - J^h$)
 - Funciones a relajar (f_i con $i \in J^h$), con las cantidades máximas a relajar (Δf_i^h con $i \in J^h$)

De esta manera se tiene la región factible de la siguiente iteración:

$$X^h = \left\{ \mathbf{x} \in X / \begin{array}{ll} f_i(\mathbf{x}) \leq f_i(\mathbf{x}^{h-1}) + \Delta f_i^h & i \in J^h \\ f_i(\mathbf{x}) \leq f_i(\mathbf{x}^{h-1}) & i \in \{1, \dots, m\} - J^h \end{array} \right\}$$

- Se vuelve a calcular los pesos de la distancia de la métrica de *Tchebychev* (volver a paso 3), obsérvese que $\mathbf{x}^{h-1} \in X^h$.

Si se desea profundizar más acerca de esta metodología se puede consultar Benayoun *et al.* (1971) o Caballero *et al.* (2002).

2.3.2.2 Método Visual Interactive Approach – VIA

Este método fue publicado por Korhonen y Laakso (1986), utiliza una función normalizada de logro, la cual debe ser minimizada según el punto de referencia proporcionado por el decisor en cada iteración. Además de definir el punto de referencia, el decisor debe escoger entre varias soluciones (Caballero *et al.*, 2002).

El método se describe a continuación (Caballero *et al.*, 2002):

- Sea $h = 0$. El decisor establece los pesos (w_1, \dots, w_m) para la función normalizada de logro. Se obtiene una solución inicial factible mediante la resolución del problema:

$$\begin{aligned} & \min_{\mathbf{x}, \alpha} \alpha \\ \text{s.a. : } & w_i (f_i(\mathbf{x}) - f_i^*) \leq \alpha \quad 1 \leq i \leq m \\ & \mathbf{x} \in X \end{aligned}$$

Donde f_i^* es el punto ideal o de referencia.

- Se le pregunta al decisor si está satisfecho con la solución obtenida, si la respuesta es positiva, entonces termina el proceso y la solución final es $(\mathbf{x}^h, \mathbf{f}^h)$. Y si la respuesta es negativa, entonces $h = h + 1$ y se continúa.
- Se le pide al decisor que especifique unas nuevas metas o punto de referencia $\bar{\mathbf{q}}^h$, teniendo en cuenta los valores obtenidos en las funciones objetivo.
- Para distintos valores de λ en el intervalo $[0,1]$, incluido $\lambda = 0$, se resuelve el problema:

$$\begin{aligned} & \min_{\mathbf{x}, \alpha} \alpha \\ \text{s.a. : } & w_i (f_i(\mathbf{x}) - (\bar{q}_i^h + \lambda(f_i^{h-1} - \bar{q}_i^h))) \leq \alpha \quad 1 \leq i \leq m \\ & \mathbf{x} \in X \end{aligned}$$

Obteniendo los vectores criterio solución que se denotan por $\{\mathbf{f}^{h,0}, \mathbf{f}^{h,\lambda_1}, \dots, \mathbf{f}^{h,\lambda_s}\}$

- De entre todos estos vectores criterio $\{\mathbf{f}^{h,0}, \mathbf{f}^{h,\lambda_1}, \dots, \mathbf{f}^{h,\lambda_s}\}$, el decisor elige el más preferido, denotado por \mathbf{f}^h y a su correspondiente vector en el espacio de decisión por \mathbf{x}^h . Y se vuelve al paso 2.

Para ver un ejemplo práctico de esta metodología se puede consultar Caballero *et al.* (2002).

2.3.2.3 Método Ecuador

Es un método propuesto por Jaramillo (1999) e implementado en un sistema soporte de decisión. Este método consiste en refinar una solución inicial, obtenida mediante otros métodos de análisis multiobjetivo sin necesidad de expresar las preferencias del decisor. El decisor hace esta refinación, conociendo de antemano las consecuencias de sus posibles acciones, y después de ofrecerle información que le permita entender las relaciones en el espacio de decisión del problema, tales como las limitaciones, factibilidades, intercambios eficientes, etc. (Smith et al., 2000).

A continuación se hace una descripción de cómo funciona esta metodología (Smith et al., 2000):

- Calcula una solución inicial por el método y pesos que el decisor elija (programación de compromiso con la métrica uno, dos e infinito y programación por metas). La solución dada se muestra al decisor por medio de una gráfica en la que presenta barras verticales desplazables, una por cada objetivo, que describen para cada objetivo el mejor valor posible (independiente de los otros objetivos), el peor valor posible y el valor de la alternativa eficiente obtenida inicialmente.
- El decisor tiene la opción de quedarse con esta solución o registrarla para análisis posteriores.
- El decisor decide qué objetivo desea mejorar a expensas de algún otro y señala hasta donde está dispuesto a empeorar éste último.
- El programa inmediatamente, en cuánto le es posible, mejora el objetivo que eligió y muestra la solución eficiente y factible que cumple con estas condiciones. La solución se obtiene resolviendo el siguiente problema de programación matemática:

$$\begin{aligned} & \max f_k(x) \\ & \text{s.a. : } f_j(x) \geq \epsilon_j \\ & f_i(x) \geq \epsilon'_i \quad \forall i \neq k, i \neq j, i = 1, 2, \dots, m \\ & x \in X \end{aligned}$$

Donde k es el objetivo elegido para mejorar, j es el objetivo elegido para sacrificar hasta un nivel ϵ_j y ϵ'_i representa los valores de $f_i(x)$ de la solución anterior del ecualizador, correspondientes a los niveles marcados en las barras desplazables de cada uno de los objetivos i antes de realizar la optimización.

- El decisor tiene la opción de registrar esta solución y continuar con su búsqueda e ir registrando las soluciones que le parezcan interesantes, o si desea se puede quedar con el grupo hallado hasta aquí y continuar con el siguiente paso.
- El usuario evalúa si el grupo guardado de alternativas aún le es interesante o si por el contrario hay algunas alternativas que ya perdieron su interés y por tanto las elimina del registro. Si al eliminar estas alternativas aún queda un grupo de ellas, estas se evalúan con algún método discreto propuesto por la herramienta.

Para profundizar más en esta metodología se puede remitir a Jaramillo (1999).

2.3.3 Técnicas A-Posteriori

Este grupo de técnicas no requieren que el decisor exprese sus preferencias antes, ni durante del algoritmo, ellas se limitan a hallar la frontera de Pareto o soluciones no dominadas modificando algunos de sus parámetros.

Cada metodología se plantea con el fin de hallar todo el frente de Pareto modificando algunos de sus parámetros sin necesidad de expresar sus preferencias, y sólo hasta el final, el decisor podría escoger con cuál de las soluciones halladas se sentiría más satisfecho de acuerdo a sus prioridades, por tanto esto implicaría que antes de tomar una decisión, cada algoritmo se debe correr tantas veces sea posible como puntos de la frontera se desee encontrar.

Dentro de este grupo se podrían mencionar algunas técnicas como:

2.3.3.1 Métodos de ponderación.

Es un método propuesto por Zadeh (1963) y consiste en obtener para un problema continuo un conjunto de alternativas no dominadas, sin establecer un orden de preferencias. Zadeh demostró que optimizando la suma ponderada de las funciones objetivos se genera un punto no dominado, es por esto que plantea convertir un problema multiobjetivo en uno monoobjetivo, de la siguiente manera:

$$Max \left\{ w_1 \frac{f_1(x)}{f_1'} + w_2 \frac{f_2(x)}{f_2'} \dots + w_i \frac{f_i(x)}{f_i'} \dots + w_m \frac{f_m(x)}{f_m'} \right\}$$

Sujeto a $x \in X$

Las restricciones son las mismas del problema original. f_i' es la constante que normaliza el objetivo i con el fin de poder sumarlos entre si y w_i es el factor ponderante del objetivo i . Para cada una de las combinaciones posibles de w_i debe cumplirse que:

$$\sum_{i=1}^m w_i = 1$$

$$0 \leq w_i \leq 1$$

Con una combinación de w_i se obtiene una solución no dominada, para obtener otra solución se debe variar esta combinación.

Se puede consultar más acerca de esta metodología en Zadeh (1963) o en Smith *et al.* (2000).

2.3.3.2 Método de las restricciones.

Este método fue propuesto por Marglin (1967), y propone convertir un problema multiobjetivo en uno mono-objetivo escogiendo una de las funciones objetivo para ser optimizada y las otras se manejan como restricciones. Así:

$$\begin{aligned} & \text{Max } f_k(x) \\ \text{s.a. } & f_i(x) \geq \varepsilon_i \quad i \neq k \\ & x \in X \end{aligned}$$

Donde $f_k(x)$ se escoge arbitrariamente como la función objetivo que se desea optimizar. Se debe hacer una correcta variación de los ε_i con el fin de obtener un número adecuado de soluciones no dominadas, además de conocer los límites mínimos y máximos para cada objetivo y así saber entre que valores se debe variar ε_i

Bajo este mismo concepto, varios investigadores han propuesto diferentes algoritmos que lleven inicialmente a hallar el rango permitido de los ε_i y luego generar el conjunto de soluciones no dominadas del problema en estudio.

Para profundizar más acerca de esta metodología se puede consultar Marglin (1967) o Smith *et al.*(2000).

2.4 PROBLEMAS MULTI OBJETIVO DIFÍCILES DE RESOLVER

Dentro de los problemas reales que consideran múltiples objetivos, hay un conjunto que presentan dificultades a la hora de resolver, independientemente del enfoque que se utilice para resolverlos.

Unos de estos son los problemas no lineales y los combinatoriales. Los primeros, son aquellos donde sus funciones objetivos o las restricciones no pueden ser representadas de forma lineal, sino por el contrario son representadas por funciones no lineales y que por lo general son muy difíciles de resolver con una técnica clásica. Los segundos son aquellos que cuentan con muchas variables de decisión y estas toman valores de 0 ó 1.

En esta investigación, tomaremos un problema del segundo tipo.

2.4.1 Problemas no Lineales

Existen problemas reales tanto monobjetivo como multiobjetivo, donde sus objetivos y restricciones no es posible aproximarlos a funciones lineales, por tanto es necesario hacer uso de ecuaciones no lineales y resolverlos por medio de modelos de programación no lineal.

Este tipo de programación tiene complejidades como el tener que considerar que ahora la solución no necesariamente la encontrará en los extremos de la región factible como es el caso de la programación lineal, y por tanto debe considerar todas las soluciones dentro de esta región. Además de ello, es posible hallar una solución como óptima, pero que en

realidad es local y no global, por lo cual los algoritmos de programación no lineal deben cuidarse de no quedarse atrapados en una solución de estas (Hillier y Hillier, 2002).

2.4.2 Problemas combinatoriales

Existen casos, en la programación monoobjetivo y multiobjetivo, en que aunque el espacio de las variables de decisión sea discreto, no existe un pequeño grupo de alternativas posibles, sino que éste es bastante numeroso o inclusive infinito, cuando esto ocurre se tiene un problema de optimización definido como combinatorial. En él las opciones suelen ser todas las posibles combinaciones de todos los posibles valores de las variables de decisión que cumplan ciertas restricciones (Smith et al., 2000). Algunos ejemplos típicos combinatoriales son la programación de tareas (proceso de producción de una fábrica), selección de rutas a seguir (problema del viajero), problemas de ensamble secuencial (orden a seguir para colocar las piezas en la fabricación de un automóvil), selección de proyectos, entre otros.

Este tipo de problemas, por lo general son *NP-Hard*, es decir, el tiempo utilizado por el algoritmo para resolverlo se incrementa considerablemente con sólo aumentar en una unidad el número de variables (Smith et al., 2000), y se hacen difíciles de abordar con las técnicas clásicas de programación matemática.

Para este tipo de situaciones y para los casos en que el problema que se quiere optimizar es matemáticamente complejo con funciones no lineales, se ha desarrollado otro tipo de métodos conocidos como meta-heurísticos.

2.5 TÉCNICAS META-HEURÍSTICAS

Estos métodos, también llamados sistemas inteligentes por autores que vienen del área de Inteligencia Artificial, consisten en sistematizar ideas para desarrollar algoritmos eficientes que encuentran "buenas soluciones" a problemas de optimización monoobjetivo; estas soluciones, en muchos casos son aproximadas a la solución óptima. Las técnicas meta-heurísticas son útiles cuando se desean resolver problemas cuyo modelo matemático no puede ser formulado fácilmente o cuando tienen espacios de búsqueda muy grandes. Las mismas combinan la simplicidad de sus ideas con su gran eficiencia para obtener muy buenas soluciones para este tipo de problemas.

Desde los últimos 20 años han sido propuestas varias técnicas generales para diseñar estos algoritmos heurísticos. Sus ideas están, en muchos casos, originalmente basadas en usar modelos provenientes de la física, la biología o la genética, entre otros.

Entre las más conocidas están los algoritmos evolutivos (*Evolutionary algorithms*), la búsqueda tabú (*Tabu search*), colonia de hormigas (*Ant colony*), nubes de partículas (*Particle swarm optimization*) y enfriamiento simulado (*Simulated annealing*). Es de

interés para esta investigación estudiar los Algoritmos evolutivos, en especial, los algoritmos genéticos, los cuales están basados en el mecanismo de selección natural de los seres vivos; por tanto sólo se hará referencia a estos en la próxima sección.

2.6 ALGORITMOS EVOLUTIVOS O COMPUTACIÓN EVOLUTIVA

Recientemente, los algoritmos evolutivos se han encontrado muy útiles para resolver problemas multiobjetivo, puesto que pueden lidiar simultáneamente con un conjunto de soluciones posibles (denominada “población”), y de esta forma permitir encontrar varios individuos del conjunto de óptimos de Pareto en una sola ejecución, contrario al caso de las técnicas clásicas de programación matemática (Coello, 2005).

Adicionalmente los algoritmos evolutivos son menos susceptibles a algunas características de la frontera de Pareto como concavidad, convexidad y discontinuidad, las cuales son consideradas como una dificultad para las metodologías tradicionales (Abbass et al., 2001).

Otras de las ventajas que se pueden identificar en este tipo de algoritmos, es que dada su naturaleza le permiten hibridarse fácilmente con otras técnicas de búsqueda u optimización al igual que explotar las arquitecturas en paralelo. Además de esto, son técnicas robustas ante cambios dinámicos y sus parámetros pueden auto-adaptarse. En cuanto a su aplicabilidad, estas tiene un campo bastante amplio y pueden llegar a ser superiores a las técnicas tradicionales, ya que son capaces de resolver problemas que aún no se le conoce solución (Coello, 2004).

Es posible identificar, dentro de los algoritmos evolutivos o computación evolutiva, tres paradigmas principales como lo son la Programación evolutiva, Estrategia Evolutiva y los Algoritmos Genéticos (Coello, 2007). A continuación se hace una breve descripción de cada uno de ellos.

2.6.1 Programación Evolutiva

Fue inicialmente propuesta por Fogel *et al.* (1966). La propone como una metodología que ejemplifica la evolución al nivel de las especies. Esta hace énfasis en los nexos de comportamientos entre padres e hijos.

El algoritmo básicamente consiste en generar aleatoriamente una población inicial, aplicar mutación y calcular la calidad de los hijos que definirá la selección por torneo entre ellos para escoger las soluciones que se retendrán para la próxima iteración. Este algoritmo a diferencia de los otros dos enfoques, no usa el operador de recombinación o cruzamiento (Santana y Coello, 2006).

2.6.2 Estrategia Evolutiva

Peter Bienert, Ingo Rechenberg y Hans-Paul Schwefel (1965) desarrollaron una metodología para resolver problemas hidrodinámicos de alto grado de complejidad, esta metodología fue inspirada en el mecanismo de mutación que ocurre en la naturaleza (Santana y Coello, 2006).

Este enfoque utilizaba una población de λ padres, del cual generaban un solo hijo, que reemplazaba el peor padre de la población. Más adelante Schwefel propone dos mecanismos de selección, uno de ellos consiste en que los μ mejores individuos obtenidos de la unión de padres e hijos sobrevivan y el otro es que sólo sobrevivan los μ mejores hijos (Santana y Coello, 2006).

En este enfoque existe el operador de recombinación, el cual puede ser sexual o panmítico. El primero de ellos actúa sobre dos individuos seleccionados aleatoriamente de la población de los padres; y el segundo se elige un padre al azar, manteniéndose fijo mientras se elige al azar un segundo padre para cada componente de sus vectores (Santana y Coello, 2006).

2.6.3 Algoritmos Genéticos

Fue planteado por John H. Holland, bajo el nombre de “*planes reproductivos genéticos*” a principios de los 1960’s (Holland, 1962). Este algoritmo usa los operadores de cruzamiento, mutación y la selección probabilística, siendo el cruzamiento el operador principal (Santana y Coello, 2006).

Para aplicar este algoritmo se requiere representar inicialmente una población aleatoria que constituya las primeras posibles soluciones y a partir de allí mediante una función que permita evaluar la “calidad” de cada solución, selecciona “individuos padres” que generen nuevos “individuos hijos”. Estos últimos se obtienen a partir de aplicar los operadores de cruzamiento y mutación a los individuos padres. Estos hijos estarían cada vez más cerca de ser los individuos pertenecientes a la frontera de Pareto.

El algoritmo requiere conocer parámetros como el tamaño de la población, probabilidad de cruzamiento y mutación, número máximo de generaciones, etc. (Santana y Coello, 2006).

Este tipo de algoritmo será desarrollado con mayor detalle en el siguiente capítulo, dado que es de interés para esta investigación, pues son reconocidos como una técnica bastante exitosa para solucionar problemas multiobjetivo de alta complejidad, y además como una técnica muy prometedora a la que se le pueden hacer algunas mejoras con el fin de aumentar su efectividad a la hora de resolver problemas como los combinatoriales.

2.7 CONCLUSIONES

Existen problemas multiobjetivo que no son fáciles de resolver mediante las técnicas clásicas de optimización, ya sea porque el problema tiene un espacio de búsqueda sumamente grande tal que las técnicas clásicas se demorarían en hallar soluciones interesantes, porque no sea fácil definir ecuaciones que representen el problema o porque estas ecuaciones sean sumamente complejas, haciendo que las técnicas clásicas no encuentren buenas soluciones.

Para este tipo de problemas se han desarrollado técnicas meta-heurísticas como los Algoritmos Evolutivos, los cuales han demostrado ser una muy buena herramienta en la solución de problemas reales. Dentro de este tipo de algoritmos pertenecen los algoritmos genéticos los cuales son reconocidos como una técnica muy buena en la solución de problemas multiobjetivo. Es este tipo de algoritmo el objeto de esta investigación, por tanto se profundizará acerca de ellos en el siguiente capítulo.

3 ALGORITMOS GENÉTICOS

3.1 INTRODUCCIÓN

Los algoritmos genéticos tienen como operadores, la selección, el cruzamiento y la mutación. En este capítulo se describirá cada uno de ellos y algunas propuestas que han desarrollado acerca de su implementación.

En la sección 3.2 se describe cómo funciona un algoritmo genético, en la sección 3.3 se describe el proceso de selección y los tipos de selección que han desarrollado, en la sección 3.4 los tipos de cruzamiento y finalmente en la sección 3.5 se describe cómo funciona el operador de mutación.

3.2 DESCRIPCIÓN DE UN ALGORITMO GENÉTICO

Este algoritmo fue desarrollado por John H. Holland (1962), el cual lo propone motivado por resolver problemas de aprendizaje de una máquina (Coello, 2004).

Este es un método de búsqueda dirigida e iterativa basada en probabilidad. Inicialmente consideraba variables de decisión que sólo podían tomar los valores de cero y uno, pero versiones posteriores del método lo han ampliado para la resolución de problemas cuyas variables de decisión pueden tomar valores diferentes a cero y uno, representándolas en código binario, tal como se muestra en el Diagrama 1.

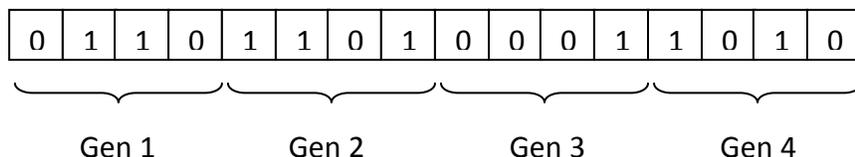


Diagrama 1. Representación de una solución en los algoritmos genéticos

Tomado de Coello (2004)

La cadena binaria completa es llamada “cromosoma” y representa una solución. A cada subsección de un cromosoma se le denomina “gen” y constituye el valor que toma una variable de decisión (Coello, 2004).

La codificación de los parámetros (por ejemplo, binaria) es conocida como “genotipo” y a la decodificación del cromosoma, se denomina “fenotipo”, es decir a los valores obtenidos al pasar de la representación (binaria) a la usada por la función objetivo (Coello, 2004).

El procedimiento de un algoritmo genético tradicional consiste en tres operaciones básicas: selección, cruzamiento y mutación, las cuales se describirán con mayor detalle más adelante.

El proceso se inicia generando, en forma aleatoria, un conjunto de vectores o cadenas (cromosoma) conocidos como individuos, que representan una muestra de la población total de las posibles soluciones. Al conjunto de estos individuos se le conoce como población. Cada uno de estos individuos tienen una característica propia y es relacionada con el valor de la función objetivo, el cual se llama “*fitness*”, “aptitud” o “calidad” de la cadena, y que de ahora en adelante se llamará aptitud (f_i).

La operación de selección escoge de la población actual (sea inicial o no) las alternativas que presenten mayor valor de aptitud para, a partir de estas, crear nuevos individuos y potencializar así sus características dentro de las nuevas poblaciones. La creación de nuevos individuos se da por medio de las operaciones de cruzamiento y mutación. El cruzamiento es un intercambio genético entre pares de individuos (llamados padres) con muy alto valor de aptitud. Y la mutación consiste en cambiar algunos valores de los genes, elegidos aleatoriamente, de los individuos obtenidos después del cruzamiento, así, si el valor del gen escogido es uno, se cambia a cero y viceversa. La mutación se realiza con el fin de evitar que se dé una convergencia prematura a óptimos locales. Los individuos productos del cruzamiento y mutación de los padres son llamados hijos y remplazan a los padres en la nueva población o “generación”.

Con el Diagrama 2, se puede tener mayor claridad acerca del proceso de un algoritmo genético. Y a continuación, basados en el manuscrito elaborado por Coello (2004), se describirán las operaciones de selección, cruzamiento y mutación; y los diferentes algoritmos que han propuesto para desarrollarlas.

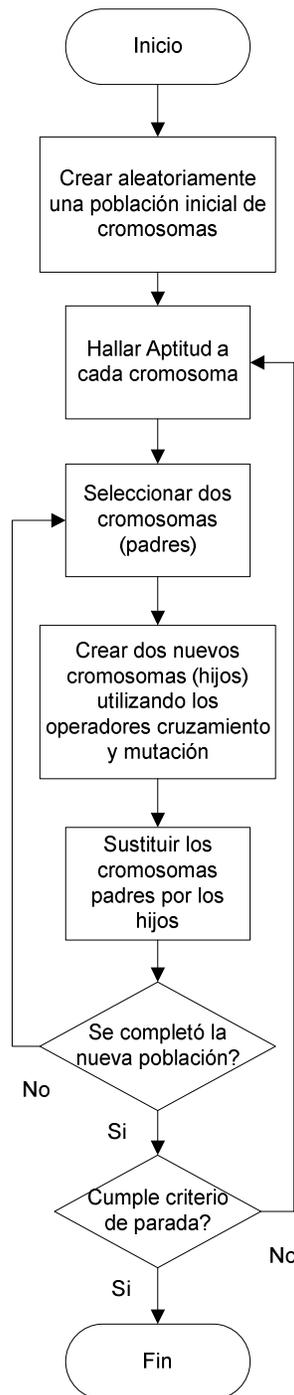


Diagrama 2. Representación del proceso de un Algoritmo Genético

3.3 SELECCIÓN

Esta es una parte importante dentro de los algoritmos genéticos y a diferencia de las estrategias evolutivas, este es un proceso probabilístico, es decir, los individuos con menor valor de aptitud, aún tienen probabilidad de reproducirse. Existen diversas técnicas de selección que pueden ser utilizadas en un algoritmo genético, a continuación describiremos algunas de ellas, clasificándolas en selección proporcional, selección mediante torneo y selección de estado uniforme.

3.3.1 Selección Proporcional

Fue propuesta inicialmente por Holland (1975) y plantean elegir los individuos según su valor de aptitud con respecto al total de la población. Dentro de este tipo de selección se pueden mencionar las siguientes técnicas, las cuales se explicarán mediante un ejemplo.

Ejemplo: Se tiene una población de cuatro individuos, con la siguiente representación binaria y el siguiente valor de aptitud.

	Cadena	Aptitud
1	110100	220
2	011010	140
3	111001	315
4	001101	42

3.3.1.1 Ruleta

Fue propuesto por De Jong (1975) en su tesis doctoral y su uso en los algoritmos genéticos ha sido muy común. Esta técnica consiste en los siguientes pasos:

- Sumar el valor de aptitud entre todos los individuos:

$$\sum_i^{TamPob} f_i = 220 + 140 + 315 + 42 = 717$$

- Hallar la probabilidad de selección para cada individuo:

Para el individuo 1:

$$P_{Selección_1} = \frac{f_1}{\sum f_i} = \frac{220}{717} = 0.31$$

Para todos los individuos

	Aptitud	Prob de Selección
1	220	0.31
2	140	0.19
3	315	0.44
4	42	0.06

- Generar un número aleatorio r donde a $r \in [0,1]$

$$r = 0.6$$

- Recorrer los individuos de la población sumando la probabilidad de selección hasta que la suma sea mayor o igual a r

(1) suma = 0.31 < 0.6

(2) suma = 0.50 < 0.6

(3) suma = 0.94 > 0.6 luego seleccionamos el individuo 3 como padre.

3.3.1.2 Sobrante Estocástico

Fue propuesto por Booker y Brindle. Tomando el ejemplo anterior, y normalizando el valor de aptitud de cada individuo mediante f_i / F donde F es la aptitud promedio de todas las cadenas, se tiene:

	Aptitud	Aptitud normalizada	Enteros	Diferencia
1	220	$220/179.25 = 1.23$	1	0.23
2	140	$140/179.25 = 0.78$	0	0.78
3	315	$315/179.25 = 1.76$	1	0.76
4	42	$42/179.25 = 0.23$	0	0.23
Promedio (A)	179.25	-	-	-

Esta técnica consiste en los siguientes pasos:

- Se selecciona como padres, de manera determinística, los individuos que tengan partes enteras.

Padres: (1) y (3)

- Si aún no se ha completado la población de padres, se completa según los valores restantes (diferencia). Para completar la población existen dos variantes:

Sin Remplazo: Consiste en usar el restante para sesgar el tiro de una moneda y saber si un individuo es nuevamente seleccionado como padre o no.

Tiro (1)= 0.23
 Tiro (2)= 0.78
 Tiro (3)= 0.76
 Tiro (4)= 0.23

Se procede hasta completar la población de padres requeridos.

Con Remplazo: Los sobrantes se utilizan para dimensionar los segmentos de una ruleta y luego se escogen los individuos de manera tradicional, así:

	Diferencia	% del Total	
1	0.23	0.115	11.5%
2	0.78	0.39	39%
3	0.76	0.38	38%
4	0.23	0.115	11.5%
	$\sum = 2$	$\sum 1$	

$$\text{Donde } \% \text{ del Total} = \text{Diferencia}_i / \sum_i \text{Diferencia}_i$$

3.3.1.3 Universal Estocástica

Fue propuesta por Baker con el objetivo de minimizar la mala distribución de los individuos. Consiste en los siguientes pasos:

- Calcular la aptitud normalizada, tal como se describió en el sobrante estocástico.
- Generar un número aleatorio entre 0 y 1. Por ejemplo, $r = 0.5$
- Comprobar si la aptitud normalizada del primer individuo es mayor que el número aleatorio, de serlo se escoge el individuo como padre. Por tanto, aquellos individuos que tengan su valor de aptitud normalizada mayor que 1, siempre serán seleccionados. Así:

(1) $1.23 > 0.5$ El individuo (1) es seleccionado como padre.

- Luego de seleccionar un individuo como padre, se le suma 1 al número aleatorio ($r = 1.5$) y ahora se comprobará este nuevo número con el valor de aptitud del individuo que está evaluando. Así:

(1) $1.23 < 1.5$ El individuo (1) no se vuelve a seleccionar como padre.

- Para evaluar el siguiente individuo, se acumula el valor de aptitud anterior con el del individuo actual y se compara con el nuevo valor de r , para el ejemplo se tiene:

(2) $2.01 > 1.5$ El individuo (2) es seleccionado como padre.

- Se continúa el proceso hasta seleccionar el número de padres necesarios, así:

($r = 2.5$)

(2) $2.01 < 2.5$ El individuo (2) no se vuelve a seleccionar como padre.

(3) $3.77 > 2.5$ El individuo (3) es seleccionado como padre.

($r = 3.5$)

(3) $3.77 > 3.5$ El individuo (3) se vuelve a seleccionar como padre.

Por tanto, los padres seleccionados en el ejemplo serían los individuos: 1, 2, 3, 3

3.3.1.4 Muestreo Determinístico

Es similar al sobrante estocástico, pero requiere un algoritmo de ordenación. Consiste en:

- Calcular la actitud normalizada, tal como lo hace el sobrante estocástico.

	Aptitud	Aptitud Normalizada	Enteros	Fracciones
(1)	220	1.23	1	0.23
(2)	140	0.78	0	0.78
(3)	315	1.76	1	0.76
(4)	42	0.23	0	0.23
	$\Sigma = 717$	$\Sigma = 4$	$\Sigma = 2$	

- Se selecciona como padre, determinísticamente, los individuos con parte entera en su aptitud normalizada, es decir, en el ejemplo se seleccionan los individuos (1) y (3) como padres.

- Para completar el número de padres, se debe ordenar la población de acuerdo a las partes decimales de mayor a menor, y se escogen los primeros de la lista.

Individuo	Fracciones Ordenadas
(2)	0.78
(3)	0.76
(1)	0.23
(4)	0.23

Luego los padres que completan la lista son (2) y (3).

3.3.1.5 Selección por Jerarquías

Fue propuesto por Baker con el fin de evitar la convergencia prematura en las técnicas de selección proporcional. Consiste en seleccionar los individuos con base en su rango y no con base en su aptitud. El algoritmo es el siguiente:

- Ordenar la población con base a su aptitud, siendo 1 el menos apto.

	Aptitud	Ordenamiento
(1)	220	3
(2)	140	2
(3)	315	4
(4)	42	1

- Hallar el valor de evaluación para cada individuo mediante la siguiente ecuación:

$$Vlr Eval(i, t) = Min + (Max - Min) \frac{jerarquia(i, t) - 1}{N - 1}$$

Donde, N es el número de individuos, Max es un parámetro entre 1 y 2 y $Min = 2 - Max$. Baker recomienda que $Max = 1.1$

Para el individuo 1:

$$Vlr Eval(1, t) = 0,9 + (0,2) \frac{3 - 1}{4 - 1} = 1,0333$$

Para todos los individuos se tiene:

	Aptitud	Jerarquía	Vlr Evaluación
1	220	3	1,033

2	140	2	0,967
3	315	4	1,1
4	42	1	0,9

- Con este valor de evaluación se aplica la selección proporcional para hallar los padres.

3.3.2 Selección mediante Torneo

Esta técnica fue propuesta inicialmente por Wetzel y consiste en hacer la selección dependiendo la comparación directa entre individuos. De ésta técnica se tienen dos enfoques:

3.3.2.1 Determinístico

Continuando con el ejemplo:

- Barajar los individuos de la población.

	Aptitud	Barajar
1	220	3
2	140	1
3	315	4
4	42	2

- Escoger un número de individuos p para competir por ser padres, por lo general son 2 individuos.

(3) y (1)

(4) y (2)

- Comparar estos individuos con base en su valor de aptitud y se escoge el individuo que tenga mayor valor de aptitud como ganador del torneo y como primer padre. Para el ejemplo:

3 y 1: El ganador como primer padre es el individuo (3)

4 y 2: El ganador como primer padre es el individuo (2)

- Se baraja la población p veces para seleccionar N padres, donde N es el tamaño de la población.

En el ejemplo, dado que p es 2, se debe barajar la población dos veces.

	Aptitud	Barajar
1	220	3
2	140	2
3	315	4
4	42	1

(3) y (2)

(4) y (1)

Los ganadores como segundo padre son los individuos (3) y (1).

Luego los padres serían: (3) y (3); (2) y (1)

3.3.2.2 Probabilístico

Funciona igual que el anterior algoritmo, sólo que se decide escoger el individuo que tiene mayor valor de aptitud con una probabilidad dada r , donde $0.5 < r \leq 1$

3.3.3 Selección de Estado Uniforme

Ésta técnica fue propuesta por Whitley, consiste en reemplazar los μ individuos peores de una generación por los μ individuos mejores de la siguiente generación. A continuación se describe el algoritmo:

- Se le llama G a la población original.
- Seleccionar R individuos ($1 \leq R < G$) de entre los que tienen mayor valor de aptitud, por ejemplo $R=2$.
- Ejecutar el cruzamiento y mutación a los R individuos que se han seleccionado y obtener los hijos (H).
- Elegir los μ mejores individuos de H , es decir los μ individuos que tienen mayor valor de aptitud entre los hijos.
- Reemplazar los μ peores individuos de G por los μ mejores individuos de la nueva generación (H).

3.4 CRUZAMIENTO

Este es un proceso que simula el cruzamiento entre dos cromosomas en un sistema biológico. Consiste en tomar dos cadenas de cromosomas o individuos e intercambiar segmentos de ellas. Este es el operador más importante en un algoritmo genético y

trabaja como un operador de explotación, es decir, cruza los individuos con el fin de concentrar su búsqueda en la misma zona. Existen tres técnicas básicas de cruzamiento:

3.4.1 Cruce de un punto

Esta técnica fue propuesta por Holland en 1975. Y consiste en escoger un punto de la cadena del cromosoma y combinar los segmentos entre dos individuos.

Este tipo de cruzamiento tiene una deficiencia y es que considera que los bloques son pequeños, lo cual es contrario en los problemas de la vida real donde la cadena de cromosoma es bastante grande y por tanto, los bloques que se combina también lo son, es por esto que en muchas oportunidades suele no presentar resultados apropiados (Coello, 2004).

El Diagrama 3 describe el cruzamiento en un punto.

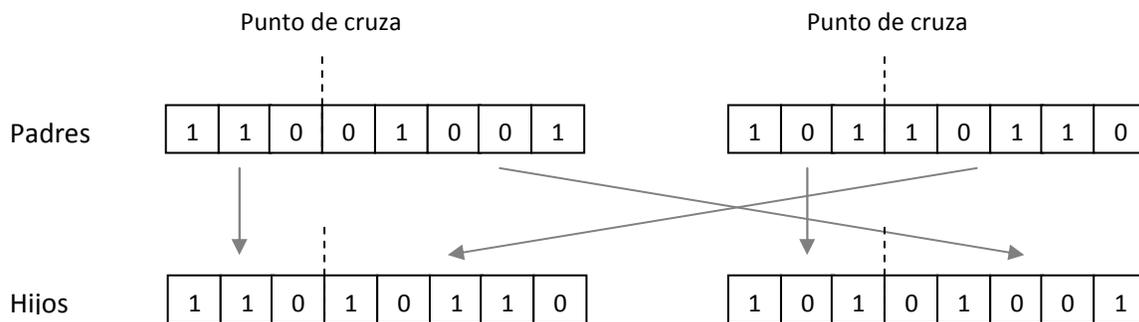


Diagrama 3. Representación de Cruzamiento en un punto

3.4.2 Cruce de dos puntos

Fue usado por primera vez por DeJong (1975). El cromosoma es partido en tres segmentos, de los cuales dos de ellos son cruzados y uno permanece constante. El Diagrama 4 representa este tipo de cruzamiento.

3.4.3 Cruce uniforme

Esta técnica fue propuesta inicialmente por Ackley (1987). Consiste en hacer un cruce de n puntos, solo que los puntos no se conocen con anticipación. Esta es la que produce mayor ruptura brusca entre las tres técnicas, lo cual debe controlarse usando una baja probabilidad de cruzamiento ($P_c < 0.5$). En el Diagrama 5 se puede ver cómo funciona este tipo de cruzamiento.

Si se desea profundizar más acerca de tipos de cruzamiento, se puede consultar Coello (2004).

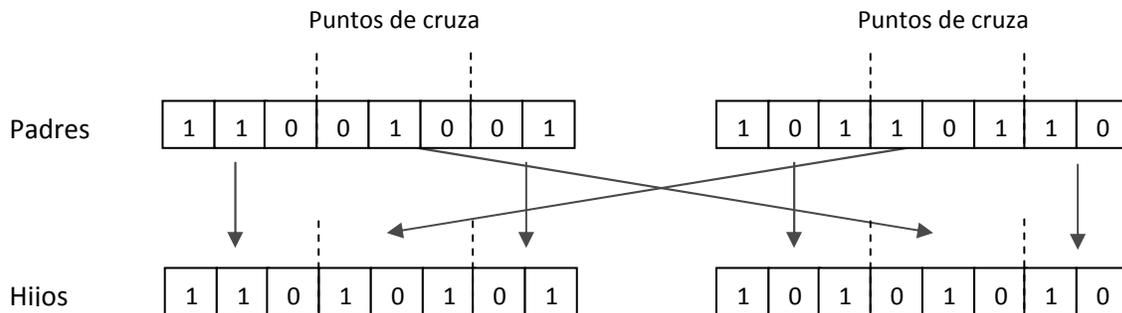


Diagrama 4. Representación de Cruzamiento en dos puntos

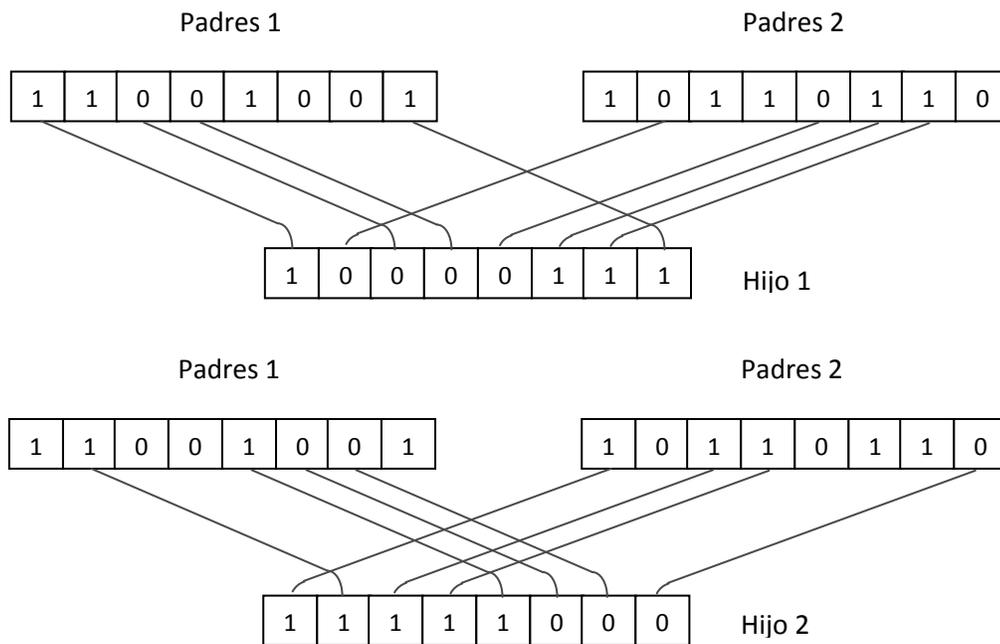


Diagrama 5. Representación de Cruzamiento Uniforme
 Tomado de Coello (2004)

3.5 MUTACIÓN

En los algoritmos genéticos, este operador se utiliza con menor frecuencia en comparación con el operador de cruzamiento; es un operador de exploración, puesto que le permite al algoritmo explorar en otras regiones, ayudándole muchas veces a salir de óptimos locales.

Este operador se lleva a cabo con una probabilidad no muy alta, en la práctica se utilizan entre 0.001 y 0.01 para una representación binaria. Sin embargo, algunos investigadores sugieren utilizar probabilidades de mutaciones altas al inicio del proceso y en la medida que avance el número de generaciones descender la probabilidad de mutación. Otros investigadores sugieren que la probabilidad de mutación sea $P_m = \frac{1}{TamCadena}$ como mínimo (Coello, 2004).

Para problemas donde las variables de decisión toman valores de 0 y 1, el operador de mutación funciona tal como se describe en el siguiente algoritmo y se representa en el Diagrama 6.

- El decisor debe determinar la probabilidad de mutación, por ejemplo $P_m = 0.009$
- Hallar un número aleatorio entre 0 y 1 por cada gen, por ejemplo $R_1 = 0.2$
- Determinar si se debe mutar o no el valor de la variable. Si $R_1 < P_m$ se muta la variable, de lo contrario no se mutaría.

$$R_1 = 0.2 > 0.009 = P_m \quad \Rightarrow \quad \text{El gen 1 no se muta}$$

- El proceso se repite para cada variable.

$$R_6 = 0.002 < 0.009 = P_m \quad \Rightarrow \quad \text{El gen 6 se muta}$$

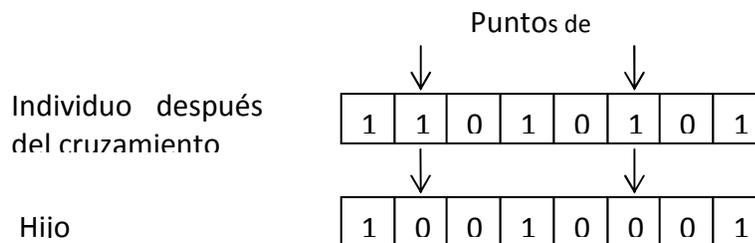


Diagrama 6. Representación de Mutación

3.6 CONCLUSIONES

En la actualidad han desarrollado diversas técnicas para la selección, cruzamiento y mutación de un algoritmo genético, las cuales son usadas tanto para resolver problemas monobjetivo como multiobjetivo. El determinar cuál de ellas usar dependerá de quien esté haciendo uso de esta metodología.

Teniendo claro cómo funciona un algoritmo genético, en el siguiente capítulo se describirá el uso de esta metodología para resolver problemas multiobjetivo, para ello se describen los principales algoritmos que han hecho historia en el campo de los algoritmos genéticos multiobjetivo.

4 ALGORITMOS GENÉTICOS PARA RESOLVER PROBLEMAS DE OPTIMIZACIÓN MULTI OBJETIVO

4.1 INTRODUCCIÓN

En las últimas décadas se ha incrementado el uso de los algoritmos genéticos para resolver problemas multiobjetivo, gracias a su fortaleza de poder trabajar simultáneamente con un conjunto de posibles soluciones cada vez más eficientes y poder obtener resultados satisfactorios.

Debido a la importancia que han cobrado los algoritmos genéticos, algunos investigadores han propuesto diferentes mejoras en la selección, cruzamiento y mutación con el fin de obtener soluciones más cercanas a la frontera real de Pareto y mejor distribuidos a lo largo de ésta. En la sección 4.2 se describirán brevemente algunos de estos algoritmos que han marcado la historia durante los últimos años en la literatura de los algoritmos genéticos multiobjetivo y en la sección 4.3 se harán algunas observaciones sobre estos.

Dado que es importante que el decisor siempre involucre sus preferencias a la hora de resolver problemas multiobjetivo, los algoritmos genéticos también se han adaptado para que cumplan con este principio, algunos de estos algoritmos se mencionan en la sección 4.4 y algunas de sus debilidades y fortalezas en la sección 4.5, para finalmente en la sección 4.6 mencionar las características que se tendrían en cuenta en una nueva metodología.

4.2 ALGORITMOS GENÉTICOS MULTI OBJETIVO

En el paso de los años, autores han propuesto diversos algoritmos genéticos para problemas multiobjetivo buscando siempre obtener un frente de Pareto mejor delineado y con mejores soluciones. En la actualidad, se ha sumado a éste objetivo la eficiencia del algoritmo, es decir que la solución que arroje sea en el menor tiempo posible.

La primera vez que se mencionan el uso de los algoritmos genéticos para resolver problemas multiobjetivo es en 1967, por el investigador Rosenberg, el cual lo menciona en su tesis doctoral para solucionar un problema biobjetivo de química, más no implementa el algoritmo, sólo expone el problema de una manera modificada, tal que lo convierte en

un problema de un solo objetivo y lo soluciona con un algoritmo genético mono-objetivo (Coello, 2005).

A partir de allí se han desarrollado diferentes metodologías que apuntan a obtener muy buenas soluciones de problemas multiobjetivo. Algunas de ellas se mencionarán en el presente capítulo.

4.2.1 VEGA

Es un enfoque planteado por David Schaffer (1985), y es considerado como la primera metodología que se plantea realmente para resolver un problema multiobjetivo. El enfoque fue llamado *Vector Evaluated Genetic Algorithm (VEGA)* el cual es un algoritmo genético simple con un mecanismo de selección modificado.

Este algoritmo consiste en subdividir la población en tantos objetivos posea y cada subpoblación la evalúa con un objetivo específico, es decir, selecciona una futura subpoblación con base a un solo objetivo omitiendo el resto, lo mismo hace con el resto de subpoblaciones.

Luego mezcla todos los individuos, intentando que un individuo que fue el mejor en una subpoblación se re-combine con otro que fue mejor en otra subpoblación y se pueda obtener una mejor población. Este algoritmo presenta problemas porque el método de selección utilizado puede eliminar una solución no dominada que puede ser una muy buena solución para combinar. Funciona bien en los problemas donde no existe un conflicto entre los objetivos y donde utilizan funciones sencillas con una sola variable (Coello, 2005).

Éste método fue criticado por el investigador David Golberg en 1989, quien a partir de ésta crítica propone el uso de categorías no dominadas para mover la población hacia la frontera de Pareto, es decir, encontrar soluciones que dominen el resto de la población. Propone asignar un valor de aptitud a las alternativas con base a la optimalidad de Pareto, dando el valor de aptitud más alto a los que son no dominados, y los demás se las asigna proporcionalmente. Además de ello sugiere un tipo de técnica (llamada nicho) para el valor de aptitud tal que le permita distribuir los individuos a lo largo de la frontera de Pareto. A pesar de que Golberg no planteó un algoritmo específico, si indujo para que los trabajos que se desarrollaran a continuación tuvieran este mismo enfoque (Coello, 2005).

4.2.2 MOGA

En 1993 los investigadores Fonseca y Fleming desarrollaron un algoritmo llamado *Multiobjective Optimization Genetic Algorithm (MOGA)*.

Su aporte al algoritmo tradicional (Fonseca y Fleming, 1993) está en el cálculo de la aptitud para cada individuo, el cual depende del número de individuos que lo dominan. Cada alternativa tiene una clasificación de la siguiente manera:

$$rank(x_i, t) = 1 + p_i^{(t)}$$

Donde x_i es la alternativa i a quien se le está calculando el valor de la aptitud en la generación t , y $p_i^{(t)}$ es el número de individuos que dominan a i en esa misma generación. De esta manera, asumiendo que se optimiza un problema de maximización, los individuos no dominados ($p_i^{(t)} = 0$) tendrán una clasificación de 1 y entre mayor sea el número de individuos que dominan a x_i , más alto es el valor de clasificación para x_i .

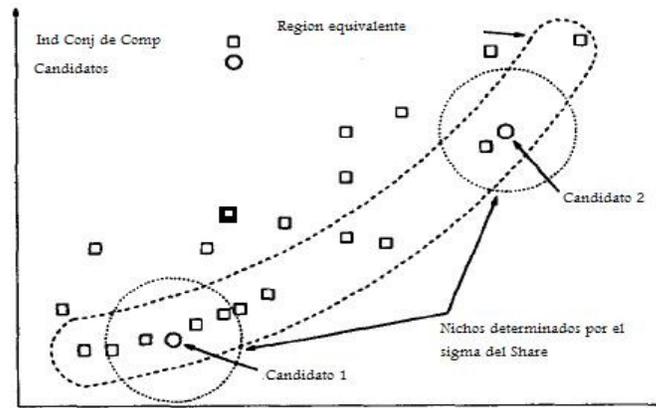
Luego la población es organizada según el orden ($rank$) de cada individuo y se asigna la aptitud interpolando desde el mejor orden ($rank = 1$) hasta el peor ($rank = n$), según una función que por lo general es lineal. Finalmente se promedia la aptitud de los individuos con la misma clasificación, entonces cada uno de los pertenecientes al mismo orden podría ser seleccionado con la misma probabilidad.

Éste fue uno de los algoritmos más efectivos en su tiempo, comparándolo con VEGA, NSGA y NPGA (Coello, 2005). Sin embargo, Goldberg y Deb señalan que el tipo de aptitud que proponen Fonseca y Fleming es posible que lleve al algoritmo a tener una convergencia prematura (Coello, 1999).

4.2.3 NPGA

En 1994 los investigadores Horn, Nafpliotis y Goldberg (1994) desarrollaron un algoritmo que llamaron *Niched Pareto Genetic Algorithm (NPGA)*, el cual incorpora los conceptos de dominancia de Pareto y de nicho en el operador de selección. El concepto de nicho es introducido con el fin de ejercer presión para extender toda la población a lo largo de toda la frontera de Pareto y se utiliza en el torneo para determinar el ganador cuando existe un empate por dominancia. Así:

Se seleccionan aleatoriamente de la población dos posibles candidatos para reproducirse, al igual que se selecciona un conjunto de individuos para comparación (por lo general es el 10% de la población total). Compite cada candidato con cada individuo del conjunto, ganando el torneo quien no sea dominado por alguno de ellos. Si los dos candidatos o ninguno de ellos son dominados por algún individuo del conjunto, entonces se escoge el ganador según el nicho en que se encuentre. En este caso gana el candidato que pertenezca al nicho que contenga menos individuos, tal como se muestra en la siguiente figura (Horn et al., 1994):



Gráfica 2. Sharing

Tomado de Horn *et al.* (1994)

En ésta selección ganaría el Candidato 2. Y el radio del nicho es fijado por el usuario como la separación mínima esperada entre las soluciones finales.

Este algoritmo a pesar de ser sencillo, es más eficiente que el MOGA, pero ligeramente menos efectivo que éste (Coello, 2002).

Si se desea investigar más acerca de este método y de cómo se calcula el nicho, se puede investigar Horn *et al* (1994).

4.2.4 NSGA

Los investigadores Deb y Srinivas (1994) crean el *Nondominated Sorting Genetic Algorithm* (NSGA). Estos toman la idea propuesta por Goldberg, que consiste en clasificar los individuos por frentes dependiendo de la dominancia.

Este se diferencia de un algoritmo genético simple por la manera de trabajar con el operador de selección. Así, para seleccionar los padres tienen en cuenta el siguiente procedimiento: buscar en la población actual quienes son no dominados y constituir el primer frente. A estos individuos se les asigna un mismo valor de aptitud *dummy* alto.

Para mantener la diversidad de la población, a los individuos del frente se les aplica un operador de distribución (*Sharing*) que intenta distribuir los puntos sobre la frontera, lo cual lo logra dividiendo el valor de aptitud *dummy* de un individuo por una cantidad proporcional al número de individuos que están a su alrededor. Como sigue:

$$f_i = \frac{f_i^*}{m_i}$$

Donde f_i^* es el valor de aptitud *dummy* y

$$m_i = \sum_j^{pob} Sh(d_{ij})$$

Esta última cantidad se calcula de acuerdo a un operador *Sharing*, así:

$$Sh(d_{ij}) = \begin{cases} \left(1 - \frac{d_{ij}}{\sigma_{Share}}\right)^2 & d_{ij} \leq \sigma_{Share} \\ 0 & d_{ij} > \sigma_{Share} \end{cases}$$

Donde σ_{Share} es el radio del nicho fijado por el usuario y que representa la separación mínima esperada entre las soluciones finales, y d_{ij} es la distancia del individuo i al individuo j .

Después de distribuirlo, estos individuos no dominados son ignorados temporalmente para hallar la siguiente frontera no dominada entre los individuos restantes. A esta última frontera se le asigna un nuevo valor de aptitud *dummy* menor que el más pequeño ya distribuido de la frontera anterior, esto se hace hasta clasificar toda la población en varias fronteras.

Terminada la clasificación, la selección la hacen teniendo en cuenta el valor de aptitud *dummy*.

Este algoritmo no fue muy eficiente porque la frontera de Pareto debe ser reparada muchas veces, por lo tanto resultó ser un algoritmo muy lento, poco efectivo e implicaba altos costos en su implementación (Coello, 2002).

4.2.5 SPEA

Los investigadores Zitzler y Thiele (1999), crearon un algoritmo llamado *Strength Pareto Evolutionary Algorithm* (SPEA) el cual utiliza un archivo externo para retener las soluciones no dominadas globales previamente encontradas; en cada generación se copia en este archivo las soluciones no dominadas y se borran las que pasan a ser dominadas.

Si el número de individuos no dominados en el archivo externo, supera el número dado permitido N' entonces, se reducen los individuos mediante *clúster*. SPEA lo hace con el método de vinculación promedio (*The average linkage method*). Este método consiste en lo siguiente:

1. Inicializar el conjunto de *clúster* C , donde cada *clúster* lo conforma un individuo del conjunto externo no dominado. $C = \bigcup_i \{i\}$

2. Si $|C| \leq N'$ entonces, pase al paso 5, sino continúe con el 3.
3. Calcule la distancia de todos los posibles pares de *clúster*. La distancia entre dos *clúster*, por ejemplo 1 y 2, está dada por la distancia promedio entre todos los posibles pares de individuos pertenecientes a los dos *clúster* así:

$$d = \frac{1}{|c_1| \cdot |c_2|} \cdot \sum_{i \in c_1, j \in c_2} \|d_{ij}\|$$

Donde $\|d_{ij}\|$ representa la distancia euclidiana entre los individuos i y j

4. Escoger los dos *clúster* con la mínima distancia y fusionarlo en un solo *clúster*. Se continúa con el paso 2.
5. Escoger un individuo representativo por *clúster*. SPEA recomienda el individuo centroide, es decir, el individuo que tenga menor distancia promedio a todos los individuos del clúster.

Para cada individuo perteneciente a este archivo externo se le computa un valor de fuerza s_i (*strength*) el cual es proporcional al número de las soluciones a las que cada individuo domina, así:

$$s_i = \frac{n}{N+1}$$

Donde n es el número de individuos de la población que son dominados por i y N es el tamaño de la población. La aptitud para cada individuo i perteneciente a este archivo externo es el mismo valor de fuerza $f_i = s_i$.

Cada individuo j perteneciente a la población actual se le calcula su aptitud dependiendo los valores de fuerza asignados a los individuos del archivo externo que lo dominan. Así:

$$f_j = 1 + \sum_{i, i > j} s_i$$

Se agrega el valor de uno para garantizar que la aptitud de los individuos del archivo externo sea mejor que el resto de la población, teniendo en cuenta que en la medida que el valor de la aptitud sea menor, el individuo tendrá más probabilidades de reproducirse.

La forma como se plantea la aptitud busca que sean preferidos los individuos más cerca de la frontera de Pareto y que menos individuos tengan a su alrededor.

Luego de la asignación de la aptitud se puede proceder a la selección, la cual se hace por torneo binario, escogiendo los candidatos para ser padres desde la unión del archivo externo con el resto de la población. Con estos padres se crean los hijos luego de aplicar los dos operadores: cruzamiento y mutación, tal como se plantean en un algoritmo genético sencillo.

Este algoritmo arrojó resultados muy por encima de las otras técnicas existentes al introducir el nuevo concepto de elitismo, lo cual lo llevó a obtener gran popularidad en el área de los algoritmos genéticos multiobjetivo.

4.2.6 NSGA-II

En el año 2000 los investigadores Deb, Pratap, Agarwal y Meyarivan desarrollaron un algoritmo llamado *Nondominated Sorting Genetic Algorithm-II* (NSGA-II) descrito en Deb *et al.* (2002) el cual usa un procedimiento rápido para organizar la población por no dominancia, un enfoque para preservar el elitismo y un operador, diferente al nicho, para dispersar los individuos en la frontera de Pareto.

Inicialmente organizan toda la población inicial aleatoria en fronteras de no dominancia, siendo la primera frontera los individuos que no son dominados dentro de la población. Para organizar el resto de la población, se olvida de las fronteras organizadas y evalúa los individuos no dominados organizándolos en una nueva frontera.

Le asigna la aptitud a cada individuo dependiendo la frontera a la que pertenezca, asignando como valor de aptitud 1 a los que pertenecen a la primera frontera, 2 a la segunda, y así a todos los individuos. Se debe tener en cuenta que éste algoritmo le da prioridad para reproducirse a los individuos que tengan menor valor de aptitud.

Luego seleccionan los padres por torneo binario y por medio de los operadores de cruzamiento y mutación crea los nuevos hijos.

En este punto usa la selección "mas", que consiste en que los hijos se juntan con los padres conformando una población R_t , la cual sería de tamaño $2N$. Por lo cual, esta población debe ser reducida de acuerdo al criterio de no dominancia tal que se obtenga la nueva población de tamaño N . Esta última población será la que se use para la selección, cruzamiento y mutación para crear una nueva generación.

El procedimiento para reducir la población $2N$ consiste en clasificar la población en varias fronteras no dominadas e ir completando la nueva población con las fronteras de mejor valor. En la última frontera aceptada, las soluciones son seleccionadas de acuerdo a un operador llamado *crowded*, el cual define el perímetro máximo de una caja contenida entre dos soluciones vecinas. Para hacer uso de este operador, primero se debe hallar la *métrica de estimación de la densidad* para cada individuo. Para calcularla se organiza la población de forma ascendente según el valor de un objetivo que toma cada individuo;

esto se hace para cada objetivo. Los individuos que están en los extremos, es decir, el que tienen el menor valor en ese objetivo y el que tiene el mayor valor, tendrán un valor de distancia de infinito. Y para el resto de los individuos i la distancia se calcula a través de la siguiente ecuación:

$$Dist_i = Dist_i + \frac{f_{i+1,j} - f_{i-1,j}}{f_j^{\max} - f_j^{\min}} \quad \forall j$$

Donde $f_{i+1,j}$ es el valor del objetivo j del individuo $i+1$ y f_j^{\max} , f_j^{\min} es el valor máximo y mínimo del objetivo j .

Hallada esta métrica, se puede utilizar el operador *Crowded*, el cual guía el proceso de selección en varios estados del algoritmo hacia una distribución uniforme de los individuos a lo largo de la frontera óptima de Pareto. Con este operador definimos que i es preferido a j , siempre y cuando i tenga una mejor clasificación en cuanto a frontera y en el caso de que i y j tengan la misma clasificación, entonces i es preferido a j cuando i tiene un mayor valor en la métrica de la estimación de la densidad.

Se debe tener en cuenta que este mismo operador es utilizado en la selección de padres, es decir, en el caso en que los candidatos en el torneo estén empatados por su valor de aptitud, entonces se escoge quien tenga un mayor valor en la métrica de la estimación de la densidad.

En la actualidad éste y el SPEA2 son quizás los mejores algoritmos, dado que encuentran soluciones muy cercanas a la frontera real de Pareto, por tanto son tomados como algoritmos de referencia para la comunidad científica y por ende para ésta investigación.

4.2.7 SPEA2

En el año 2001 Zitzler, Laumanns y Thiele hacen unas correcciones al algoritmo SPEA, al que llama *SPEA2* (Zitzler et al., 2001), y éste último se diferencia del primero por la asignación de la aptitud, la selección de los padres, el operador de truncamiento y en fijar el tamaño del archivo externo para todas sus generaciones.

- **Asignación de la Aptitud:** Para la asignación de la aptitud, a diferencia del SPEA, se tiene en cuenta tanto los individuos dominados como los que dominan y la densidad de cada individuo.

El valor de fuerza del individuo i , S_i está dado por:

$$S_i = |\{j | j \in P_t + \bar{P}_t \wedge i \succ j\}|$$

Donde j equivale al número de soluciones que i domina, P_t y \bar{P}_t son los individuos de la población y los individuos pertenecientes al archivo externo en la generación t respectivamente.

Teniendo en cuenta esta nueva definición de valor de fuerza, se calcula el valor de una aptitud cruda (sin tener en cuenta la densidad) para cada individuo R_i el cual está dado por la siguiente ecuación:

$$R_i = \sum_{j \in P_t + \bar{P}_t, j > i} s_j$$

Se debe tener en cuenta que en la medida que un individuo tenga un valor de aptitud pequeño será preferido a uno que tenga un valor más alto.

Otro componente de la aptitud de cada individuo es la medida de densidad D_i , la cual la proponen como una adaptación al método del vecino más cercano, así:

$$D_i = \frac{1}{\sigma_i^k + 2}$$

Donde σ_i^k es la distancia de i al k -ésimo elemento. Para hallar este valor se calcula la distancia de i a todos los individuos, tanto de la población como del archivo externo y luego se organiza estas distancias en una lista en orden ascendente, y se toma el valor del k -ésimo elemento como σ_i^k siendo $k = \sqrt{N + \bar{N}}$. Donde N es el tamaño de la población y \bar{N} es el tamaño del archivo externo. De esta forma el valor de aptitud para el individuo i esta dado por

$$f_i = R_i + D_i$$

- **Individuos del archivo externo:** Inicialmente se copia todos los individuos no dominados, tanto de la población como del archivo externo de la generación anterior, al archivo externo de la generación actual, una forma de identificarlos es seleccionando aquellos individuos que tenga valor de aptitud menor que uno.

Para el SPEA 2 el tamaño del archivo externo siempre es constante en el tiempo, por tanto, en el caso en que estos individuos no sumen la cantidad dada por el usuario \bar{N} , se debe completar el archivo con los mejores individuos dominados, es decir, con los que tengan menor valor de aptitud.

Y si el número hallado de no dominados es mayor que \bar{N} entonces se debe remover los individuos necesarios hasta que el tamaño del archivo sea \bar{N} . La forma como se escoge el

individuo a remover es hallando el que tenga la distancia mínima a otros individuos y este será eliminado, en el caso de que exista varios individuos con la misma distancia mínima, se escoge quien tenga la segunda menor distancia mínima.

- **La selección de los padres:** La selección se hace por torneo binario con remplazo participando solamente los individuos del archivo externo.

Con las mejoras propuestas para este algoritmo, efectivamente logra valores mucho mejores que el SPEA, y aunque no haya tenido la misma popularidad que éste último, ha sido preferido junto con NSGA-II para la optimización de problemas reales.

4.2.8 cNSGA-II

Deb y Goel (2001) proponen controlar el elitismo en el NSGA II, puesto que considera que el elitismo en los algoritmos genéticos multiobjetivo lleva a converger prematuramente a un conjunto de soluciones sub-óptimos, por tanto proponen controlarlo y a su vez mantener un balance entre los temas de exploración y explotación.

El control consiste en restringir el número de individuos que se aceptan de cada frontera para ser parte de la siguiente generación. El número de individuos que se aceptan depende de la siguiente distribución:

$$n_i = N \frac{1-r}{1-r^K} r^{i-1}$$

Donde n_i es el número máximo permitido de la frontera i , N es el tamaño de la población, K es el número de fronteras halladas y $r < 1$ es la tasa de reducción, dada por el usuario.

En el caso de que n_i sea mayor al número de individuos que tiene la frontera i , entonces se selecciona por torneo los n_i individuos de la frontera y en el caso de que n_i sea menor, entonces se aceptan todos los de esa frontera y los faltantes se escogen de la siguiente, por tanto el número de individuos de la siguiente frontera aumentaría en esa cantidad.

De esta forma se lograría mayor diversidad en la población y la posibilidad de encontrar una frontera mejor distribuida.

4.3 OBSERVACIONES SOBRE LOS MÉTODOS

Las anteriores metodologías tienen como finalidad encontrar toda la frontera de Pareto y mejor delineada, sin embargo para solucionar problemas del mundo real, es necesario tener una única solución.

Escoger ésta solución entre un grupo bastante grande de alternativas igualmente buenas es algo difícil, por tanto se continúa con un problema de decisión discreto.

Deb y Chaudhuri (2005) mencionan que adicional al problema de encontrar un conjunto de soluciones óptimas está el seleccionar científicamente entre ellas una solución particular.

Con el fin de obtener una única solución o un grupo pequeño de ellas se ha planteado el uso de las preferencias del decisor en este tipo de metodologías. En la siguiente sección se describen algunas propuestas de cómo incorporar las preferencias en estos algoritmos.

4.4 INCORPORACIÓN DE LAS PREFERENCIAS EN LOS ALGORITMOS GENÉTICOS MULTI OBJETIVO

Incorporar las preferencias en metodologías que resuelven problemas multiobjetivo es de gran ayuda para incrementar la selección, puesto que destaca soluciones que son relevantes para el decisor. Además de ello, también puede ayudar a direccionar mucho mejor la búsqueda para aquellos problemas que involucran muchos objetivos y que le es difícil hacerlo basado solo en el criterio de dominancia de Pareto (Rachmawati y Srinivasan, 2006).

Sin embargo, incorporar las preferencias tiene asociado una gran dificultad, debido a la incertidumbre que se presenta (por el desconocimiento previo del problema) en las preferencias del decisor que siempre son muy difusas. A su vez, la complejidad de la toma de decisiones hace muy difícil la formulación matemática de las preferencias (Rachmawati y Srinivasan, 2006).

Quien desarrolló el primer esquema que incorporó las preferencias del decisor en los Algoritmos Evolutivos Multiobjetivo fue el investigador Masahiro Tanaka en 1992, y a partir de allí este tema fue considerado como un tópico muy importante puesto que se ajustaba mucho a la solución de problemas de la vida real, ya que exigían obtener solo pocos puntos o uno sólo y no toda la frontera, sin embargo, fue un tema en el que pocos investigadores mostraron interés (Coello, 2005).

A partir de allí han desarrollado varias propuestas, las cuales algunas de ellas las mencionaremos a continuación haciendo uso de la clasificación hecha por Veldhuizen y Lamont (2000), quienes proponen clasificar los algoritmos genéticos multiobjetivo en a priori, interactivo y a posteriori, según el momento en que son incorporadas las preferencias.

4.4.1 Algoritmos A priori

Los algoritmos a priori formulan las preferencias antes de la optimización y por lo general son representadas por medio de parámetros que ayudan a formular una sola función objetivo mediante objetivos agregativos. Este tipo de algoritmo no permite cambios en los parámetros durante la búsqueda (Rachmawati y Srinivasan, 2006).

Este enfoque resulta muy sencillo y eficiente cuando se tiene una buena representación matemática de las preferencias del decisor y no presentan dificultades en el momento de computarlas, lo cual es difícil que ocurra. Además de ello se requiere de un muy buen conocimiento de los posibles resultados del modelo (Rachmawati y Srinivasan, 2006).

En 1998, Voget and Kolonko combina la lógica difusa con los algoritmos evolutivos, planteando un controlador difuso que regule automáticamente el proceso de selección de un algoritmo evolutivo, a través de un conjunto de metas predefinidas por el usuario que definan el comportamiento deseable de la población (Coello, 2000). Dado que estas metas son proporcionadas al inicio del proceso, este algoritmo es categorizado como a priori.

En el 2001, Branke J, Kaubler T, Schmeck H, proponen involucrar las preferencias mediante la representación de sacrificios entre objetivos. De forma a priori especifican dos conjuntos de pesos que corresponden al máximo y al mínimo sacrificio de cada uno de los objetivos con respecto a los otros, que está dispuesto a asumir el decisor. Luego utiliza cada conjunto de pesos para hallar una función de aptitud mediante una suma ponderada y dado que son dos grupos de pesos, se tendrían dos funciones de aptitud para evaluar cada alternativa. De esta manera, dominará la alternativa que al evaluarla tenga mayor valor (en el caso de maximización) en por lo menos una de estas dos funciones. Esta metodología presenta desventajas como no poder modelar la dependencia entre los objetivos o la dificultad de especificar los sacrificios mínimos y máximos cuando se tienen más de dos objetivos (Rachmawati y Srinivasan, 2006).

En el 2002, Cvetkovic y Parmee proponen una metodología que modela las preferencias mediante la importancia relativa entre los objetivos con la ayuda de variables lingüísticas. Construye una matriz de preferencias difusas para calcular los parámetros que luego son utilizados con el método de suma ponderada. La debilidad que presenta esta metodología es que es muy sensible ante los parámetros requeridos (Rachmawati y Srinivasan, 2006).

En el mismo año (2002), con el fin de apuntarle a la debilidad que presentaba la metodología propuesta por Cvetkovic y Parmee, Jin y Sendhoff usan intervalos de pesos (Rachmawati y Srinivasan, 2006), representando las preferencias mediante la importancia relativa entre los objetivos utilizando variables lingüísticas como: muy importante, medianamente importante o poco importante. Estas variables son transformadas en valores reales mediante metodologías como la suma ponderada aleatoria y la suma ponderada dinámica, las cuales son descritas en Jin y Sendhoff (2002), teniendo en cuenta que cada una pertenece a un intervalo dado previamente. Una vez convertida la matriz

en números reales, estas entradas ayudan a calcular los parámetros que serán utilizados en la suma ponderada (Jin y Sendhoff, 2002). Esta metodología presenta debilidades como no ahorrar tiempo de computo, no converger muy bien en los frentes cóncavos y no mostrar dependencia entre los objetivos, característica que es usual en los problemas de la vida real (Rachmawati y Srinivasan, 2006).

En el 2003, Tan, Khor, Lee y Sathikannan, representan las preferencias especificando múltiples metas mediante conjunciones como *AND* y *OR* y clasificando los objetivos por prioridades. De esta forma selecciona las alternativas no dominadas del frente de Pareto. Esta metodología tiene como desventaja la complejidad computacional y el no permitir la noción de compensaciones o cualquier otra relación entre objetivos (Rachmawati y Srinivasan, 2006).

En el mismo año (2003) Filomeno, Bersini y Bouillard, proponen una metodología combinando los algoritmos evolutivos con PROMETHEE II (Coelho et al., 2003) donde tienen en cuenta tanto las preferencias del decisor como las restricciones que deben cumplir. En esta metodología las restricciones se toman como una función objetivo más, que relacione la satisfacción por cumplir las restricciones. En la selección, se comparan los dos nuevos hijos con los dos padres que los crearon haciendo uso del PROMETHEE II, y escoge los dos mejores para hacer parte de la nueva generación. Los pesos con los que inicia esta metodología son dados por el decisor, pero en la medida que el proceso avanza estos se van autoajustando según el cumplimiento de las restricciones. Se debe tener en cuenta que en esta metodología los pesos son autoajustables según el cumplimiento de las restricciones, más no son modificados por tener en cuenta los cambios en las preferencias dadas inicialmente por el decisor.

Cvetkoviç y Coello en el 2004 mencionan en uno de sus artículos (Cvetkoviç y Coello, 2004) un algoritmo genético multiobjetivo co-evolutivo que consiste en optimizar m objetivos usando m algoritmos de forma paralela, tal que cada uno de ellos optimice un objetivo y durante el proceso de corrida se usa progresivamente una función de penalización, acorde a una distancia en el espacio genotípico, que guíe la solución a una zona común. En las primeras iteraciones se es un poco flexible al aceptar soluciones lejos de los estándares establecidos y en las últimas generaciones no se permite una variación más del 10% del rango de las variables. El objetivo más importante para el decisor es menos penalizado que el menos importante y de esta forma la solución obtienen buenos valores para los objetivos de mayor interés.

En el año 2006, Hisao Ishibuchi, Yusuke Nojima, Kaname Narukawa, y Tsutomu Doi, plantean las preferencias del decisor como una forma de mejorar la convergencia de las soluciones mientras se guarda la diversidad entre estas. Proponen hacer un híbrido entre dos enfoques de la optimización multiobjetivo: Toma de decisiones multicriterio (MCDM) y la Optimización evolutiva multiobjetivo (EMO). La primera con el fin de hacer una búsqueda eficiente y la segunda con el fin de darle al decisor un conjunto de soluciones y

no una sola solución. El híbrido consiste en involucrar las preferencias a priori en una función escalable que se utilizará para seleccionar los padres de la nueva generación. La forma como las involucre dependerá del criterio del decisor, es decir, cuando le interesa la importancia relativa entre los objetivos, entonces utilizará las preferencias en una función de aptitud de suma ponderada, o cuando requiere cumplir un nivel mínimo de los objetivos, reflejará las preferencias en una función que penalice la selección de padres cuando el individuo no cumpla con el mínimo requerido del objetivo, o si se tiene una solución de referencia que representa la preferencia del decisor, entonces la selección de los padres es dada por la distancia a esta solución de referencia (Ishibuchi et al., 2006).

Este, al igual que todos los métodos que le permite al decisor definir sus preferencias de forma a priori tienen dificultades, puesto que el decisor tendría que tenerlas bien definidas para el problema que está resolviendo y esto en la mayoría de las veces no es posible. Además de ello, los parámetros que representan las preferencias, algunas veces podrían ser muy optimistas, fuera de la región factible o algunas veces tan poco optimistas que se perdería de obtener mejores soluciones.

4.4.2 Algoritmos Interactivos

En el enfoque interactivo se formulan las preferencias antes de la optimización, pero son modificadas por el decisor durante el proceso de búsqueda, permitiéndole influir en la dirección de éste. Estos algoritmos son los menos comunes en la literatura.

Este enfoque tiene en cuenta el aprendizaje que va obteniendo del problema con la información que es revelada durante la búsqueda, lo cual hace que el decisor mejore su juicio a la hora de darle un valor a los parámetros (Rachmawati y Srinivasan, 2006).

Uno de los primeros algoritmos propuestos bajo este enfoque fue presentado en 1993 por Fonseca y Fleming y mencionado en Coello (2000). Ellos proponen una metodología basada en MOGA sumándole una meta de logro interactiva que permite ser modificada por el decisor a medida que tiene mayor conocimiento del problema. Recomiendan la construcción de un sistema experto a partir del conocimiento sobre las preferencias de los decisores y así automatizar las metas de logro.

Luego, en 1998 Fonseca y Fleming proponen priorizar objetivos y comparar las soluciones comenzando con el objetivo de mayor prioridad. Esta metodología presenta debilidades como alta complejidad computacional y no permite modelar la compensación o cualquier otra relación entre los objetivos (Rachmawati y Srinivasan, 2006).

Deb y Chaudhuri (2005), sugieren una herramienta que le permita al decisor hallar ya sea toda la frontera de Pareto o concentrarse sólo en una región específica. Además le da la opción al decisor de obtener soluciones “*knee*” (soluciones donde el decisor no desea moverse, ya que le implican un alto sacrificio de uno de sus objetivos por aumentar una unidad del otro) u obtener una frontera robusta.

Para esto, inicialmente la herramienta hace uso de un Algoritmo Evolutivo como el NSGA-II para encontrar una frontera inicial. Esta frontera es mejorada implementando varios conceptos; por ejemplo, para mejorar los extremos, la herramienta dispone del uso de algoritmos genéticos para un solo objetivo (correspondiente al extremo que se desea mejorar) y para mejorar el sector intermedio propone aplicar un algoritmo de búsqueda local para algunas soluciones de la frontera hallada con el NSGA-II. Estas soluciones se pueden seleccionar automáticamente tomando las que presenten menor densidad o se seleccionan según las preferencias del decisor a partir de ciertas metodologías como suma ponderada, función de utilidad, entre otras, la cual dependerá del decisor.

Si la frontera mejora, el algoritmo continúa trabajando con la frontera final, de lo contrario permanece la hallada con el NSGA-II. Finalmente intenta mejorarla nuevamente utilizando los métodos clásicos de ϵ -constraint (Deb, 2001), donde el decisor debe definir los parámetros.

El decisor decide si quedarse con esta frontera, concentrarse en una región particular o quedarse con un punto específico. Si desea concentrarse en una región vuelve a mejorar esta tal como se describió arriba. Este procedimiento lo repite las veces que sea necesario hasta obtener el punto o fracción deseada de la frontera.

Dentro de los algoritmos interactivos podríamos mencionar otras metodologías que buscan modificar los parámetros del algoritmo en la medida que este corre, tales como las presentadas en Cvetković and Coello (2004) donde mencionan que en algunas ocasiones, las preferencias no necesariamente deben estar dadas por el decisor o grupo de ellos, sino que pueden ser determinadas mecánicamente usando agentes, como es el caso en los diseños de ingeniería. A medida que corre el algoritmo las preferencias se van ajustando de acuerdo al cumplimiento o no de ciertas restricciones. Sin embargo, de esta forma las preferencias del usuario no serían representadas puesto que se buscaría modificar los parámetros tal que se pueda obtener una frontera más cerca a la real o mejor distribuida, pero no se centraría en una fracción de interés dada por el decisor.

Un algoritmo bajo el enfoque interactivo es muy atractivo, puesto que le permite al decisor cambiar de opinión en cuanto a sus preferencias en la medida que va conociendo los posibles resultados, sin embargo requieren de un mayor esfuerzo mental para el decisor al tener que ir ajustando sus preferencias, por tanto es conveniente que los algoritmos clasificados en este enfoque sean prácticos a la hora de definir la forma como se representen las preferencias del decisor.

4.4.3 Algoritmos A posteriori

En el enfoque a posteriori las preferencias son involucradas solo hasta que el algoritmo encuentra un conjunto de soluciones no dominadas. Bajo este enfoque el decisor se evita problemas de computo, sin embargo aparecen otros problemas como el de costo computacional, la dificultad de explorar el conjunto óptimo de Pareto para seleccionar

una única solución cuando se tiene un problema que considera muchos objetivos, y el poco progreso en la búsqueda del óptimo cuando se tienen un espacio de búsqueda grande, como es el caso de los problemas con muchos objetivos (Rachmawati y Srinivasan, 2006). Sumado a esto, a pesar del esfuerzo en encontrar todo el frente, es posible que finalmente le ofrezca al decisor un conjunto de alternativas que no cumpla totalmente con sus preferencias, si lo compara con la frontera real.

Los algoritmos categorizados en este enfoque son los algoritmos evolutivos descritos en la sección 4.2, los cuales encuentran una posible frontera de Pareto y a partir de allí, le dan al decisor la libertad para escoger la solución que implementaría. Para decidir qué solución implementar, el decisor puede hacer uso de conceptos como *outranking*, intercambio entre objetivos, función de utilidad, etc.

Massebeuf et al, por ejemplo, en 1999 plantean una metodología que utiliza el método VEGA para encontrar la curva de Pareto y luego clasifican las alternativas halladas en la curva mediante el criterio de *outranking*, hallando un índice de preferencias de una solución candidata frente a otra. Esta metodología presenta problemas, puesto que a medida que aumente el número de objetivos necesitaría especificar muchos parámetros, requiriendo mucho esfuerzo para su configuración. Además obtendría puntos que no son necesarios para el decisor (Rachmawati y Srinivasan, 2006).

Bajo éste enfoque el decisor se ve limitado a seleccionar la alternativa a partir de las soluciones no dominadas obtenidas con el algoritmo genético multiobjetivo que se esté utilizando, por tanto, en el caso de no obtener toda la frontera de Pareto, el decisor debe escoger entre las que la metodología le pueda ofrecer, aún cuando pueda existir una solución que mejor se ajuste a sus preferencias.

4.5 DEBILIDADES Y FORTALEZAS ENCONTRADAS EN LAS METODOLOGÍAS CON PREFERENCIAS

El análisis multiobjetivo es una visión de los problemas de optimización mucho más realista y completa, sin embargo, las complejidades que adiciona la concepción, formulación y resolución de este tipo de problemas ha hecho que ésta no sea muy aplicada.

Algunas de esas complejidades es que no existe una solución “óptima” bajo el concepto clásico de optimización, es decir, no existe una única solución que no pueda ser superada en la región factible, sino que existen infinitas soluciones eficientes u óptimo – paretianas. No obstante, el decisor necesita elegir solo una de ellas con el fin de implementarla, por tanto debe recurrir a identificar sus verdaderas preferencias; lo cual resultaría una tarea difícil y a veces hasta imposible.

Este es el caso de las soluciones obtenidas por algoritmos como el NPGA, NSGA, SPEA, NSGAI, SPEA 2, etc. vistos en la sección 4.2, quienes han demostrado gran eficiencia a la hora de encontrar la frontera de Pareto en los problemas multiobjetivo complejos, tales como los de alta no linealidad, gran dimensionalidad, problemas combinatoriales, etc. Sin embargo, estas metodologías sufren algunas falencias que serán mencionadas en la siguiente sección.

4.5.1 Falencias encontradas en la literatura

Las metodologías vistas en la sección 4.2 y que a la hora de incorporar las preferencias son utilizadas en los algoritmos a posteriori presentan algunas falencias, una de estas es que en problemas como la mochila encuentran sólo un conjunto discreto del “codo” de la frontera de Pareto, sin considerar que posiblemente la solución que cumple con las verdaderas preferencias del decisor esté por fuera de ese codo.

Hisao Ishibuchi y otros investigadores (2006) plantean que el NSGA II, siendo uno de los algoritmos más eficientes en la actualidad provee buenas soluciones candidatas cuando está resolviendo el problema de la mochila, pero cuando el decisor está interesado en la zona central de la frontera de Pareto, pero bajo el supuesto de que el decisor desear obtener un muy buen valor de uno de los objetivos, este algoritmo no puede proveerle una buena solución candidata.

Así mismo, en el caso de los algoritmos a posteriori el proceso de decisión sigue incompleto, porque aun faltaría implementar una metodología adicional de análisis multicriterio que permita incorporar las preferencias del decisor para obtener la solución última a implementar.

Este último proceso además de incrementar el costo computacional, crear la necesidad de usar otra metodología (incluyendo quizá otro software, otro aprendizaje, etc.), invertir más tiempo para obtener la respuesta al problema, crear la posibilidad de que la verdadera “solución preferida” no esté entre el conjunto discreto que arroja el algoritmo genético, etc., no libera al decisor de la tarea de tener que expresar sus preferencias.

Adicional a esto, si el conjunto discreto de posibles soluciones es muy grande, la toma de decisiones es muy compleja, y si por el contrario, es pequeño ésta puede ser muy simplista.

Sin bien es difícil que el decisor tenga claro “sus verdaderas preferencias” al final del proceso de un algoritmo genético, lo es aun mas al principio del proceso, como es el caso de los algoritmos a priori, ya que el decisor no conoce sus posibilidades, los valores máximos, o mínimos que podría obtener en los diferentes objetivos, los intercambios factibles y eficientes entre objetivos u otra información que le permita tener una idea sobre los efectos que podría tener sus preferencias en la búsqueda de una solución.

Deb y Chaudhuri (2005) manifiestan que los métodos existentes demandan una cantidad de información del problema de parte del decisor, sin proveerle al decisor alguna idea de la naturaleza del resultado de la frontera de Pareto óptima, haciendo que éste tenga que adivinar lo que está sucediendo.

Como se mencionará en la siguiente sección, es posible que estas deficiencias puedan ser solucionadas con un algoritmo interactivo, sin embargo este último le puede exigir al decisor un alto grado de esfuerzo mental a la hora de decidir sobre sus preferencias, por tanto este tipo de metodología también requieren de cuidado en el momento de definir como involucrar al decisor.

4.5.2 Fortalezas encontradas en la literatura

En la literatura también encontramos, aunque muy pocos, un número de algoritmos interactivos que buscan dar solución a estas dificultades.

Al incorporar las preferencias del decisor en el mismo proceso genético, se podría obtener una solución o un conjunto pequeño de soluciones optimo-paretianas que sean fáciles de analizar, con un costo computacional igual o a veces menor, pues el espacio de búsqueda sería reducido.

Este tipo de metodología le da la posibilidad al decisor de no definir sus preferencias definitivas al principio del proceso, sino que a medida que éste avanza, el decisor va obteniendo una información nueva que le permite aprender, analizar las posibilidades e ir aclarando y definiendo sus preferencias. Este tipo de información puede estar relacionada con las mejores soluciones encontradas hasta el momento, el valor máximo y mínimo que toman los objetivos durante el proceso, las tendencias, los intercambios posibles, etc.

Esto porque las preferencias no pueden ser independientes del problema en sí y de las posibilidades reales que se tienen, ya que el decisor, por lo general, antes de preferir un objetivo frente a otro, desea conocer qué nivel de beneficios tendrá garantizado.

Por tanto en un único proceso y con un mismo tiempo computacional el decisor puede tener la solución definitiva a implementar, o por lo menos un conjunto pequeño de ellas, el cual le será mucho más fácil de analizar.

Dado que la búsqueda será guiada por el decisor, este conjunto pequeño involucrará las alternativas de interés para él, las cuales no necesariamente conformarán la zona central de la frontera de Pareto, sino que estarán ubicadas de acuerdo a las preferencias del decisor.

4.6 CARACTERÍSTICAS PARA UNA NUEVA METODOLOGÍA

Conociendo las ventajas que puede proporcionar una metodología interactiva, pero teniendo en cuenta que es importante la forma como el decisor se involucre en ella, en esta investigación se propone hacer uso de este enfoque para plantear una nueva metodología que le proporcione al decisor un conjunto de soluciones, no tan grande tal que finalmente no tenga claridad a la hora de escoger la alternativa a implementar, ni tan pequeño tal que se sienta inseguro a la hora de tomar una decisión.

Esta nueva metodología debe proporcionar seguridad al decisor a la hora de definir sus preferencias, es decir, que éste lo puede hacer de una manera práctica y que pueda reevaluar sus preferencias cada vez que lo desee.

Se espera que el decisor al facilitar sus preferencias guíe la búsqueda del algoritmo genético hacia la región de su interés, y que a la vez estas limiten el espacio de búsqueda, tal que el costo computacional que se usaba para buscar en una zona más amplia, sea usado ahora para buscar en una zona más pequeña, por lo cual este esfuerzo adicional puede ser usado para mejorar la calidad (eficacia) de la frontera encontrada.

4.7 CONCLUSIONES

Inicialmente, la principal preocupación de los investigadores en este campo fue mejorar la técnica, tal que le pudieran ofrecer al decisor múltiples soluciones que se aproximen a la frontera real, sin embargo, se olvidaron de la importancia que tiene para el decisor conocer una buena solución pero conforme a sus preferencias con respecto al problema que esté resolviendo.

En los últimos años el tema de las preferencias ha cobrado importancia y se han incorporado a los algoritmos genéticos multiobjetivo antes (a priori), durante (interactivos) o después (a posteriori) de obtener la frontera de Pareto, primando los algoritmos que involucran sus preferencias antes de iniciar el proceso debido a su simplicidad.

No obstante, se debe resaltar que los algoritmos interactivos son bastante prometedores, puesto que le permiten al decisor ajustar sus preferencias de acuerdo a lo que esté aprendiendo del problema y concentrar la búsqueda sólo en su región de interés, lo cual podría llevar a obtener soluciones más cercanas a la frontera de Pareto en el mismo número de iteraciones, ya que se tiene un espacio de búsqueda más pequeño; además, para problemas como la mochila, podrían hallar soluciones que no eran posibles de obtener con un algoritmo a posteriori.

En el próximo capítulo se plantearán dos nuevas metodologías, una que halle el frente de Pareto y otra interactiva que haga uso de los beneficios de éste tipo de metodologías para

obtener sólo una porción de la frontera de Pareto conforme a los intereses del decisor, y quizás más cercana a la frontera real que si se utilizara un muy buen algoritmo genético multiobjetivo sin preferencias.

5 METODOLOGÍA PROPUESTA

5.1 INTRODUCCIÓN

En el presente capítulo se proponen dos metodologías para resolver problemas multiobjetivo, una de ellas se describe en la sección 5.2 y consiste en hacer uso de las fortalezas del NSGA II y el SPEA2 para proponer una nueva metodología que halle una frontera de Pareto cercana a la real. Mientras que la otra metodología, descrita en la sección 5.3, consiste en incorporar las preferencias del decisor de una forma interactiva, tal que éste pueda obtener una porción de la frontera de Pareto que sea de su interés y mucho más fácil de analizar.

5.2 ALGORITMO GENÉTICO ELITISTA MULTI OBJETIVO – AGEM

Dentro de la comunidad científica de los algoritmos genéticos multiobjetivo, el NSGA-II y el SPEA2 son reconocidos como los algoritmos de referencia, ya que obtienen soluciones muy cercanas a la frontera real de Pareto y bien distribuidas a lo largo de esta. Su eficiencia se debe a conceptos novedosos que manejan cada uno de ellos.

El NSGA-II por ejemplo, junta los padres con los hijos y escoge los mejores para conformar la nueva generación, esto hace que el algoritmo en un mismo número de iteraciones obtenga resultados más cercanos a la frontera real que sino los juntara. En el caso de SPEA2, éste mejora su efectividad creando un archivo externo que le permita guardar y actualizar con los mejores individuos de cada generación, los cuales también compiten a la hora de seleccionar los padres que serán reproducidos.

Dado que dentro de los algoritmos genéticos son dos conceptos interesantes, en esta investigación se propone hacer uso de ellos para crear una nueva metodología multiobjetivo y verificar si los resultados podrían ser más cercanos a la frontera real.

El algoritmo propuesto lo llamaremos Algoritmo Genético Elitista Multiobjetivo (AGEM) y sus características principales se describen a continuación:

5.2.1 Manejo de las restricciones

AGEM trabaja con individuos factibles, por tanto cada individuo infactible hallado dentro de una generación, al igual que los hijos infactibles generados a partir de los padres, deben ser reparados con una probabilidad del 100%. La reparación se hace con base a la técnica “Reparación de Greedy”, utilizada por Eckart Zitzler (1999) en su algoritmo multiobjetivo SPEA.

La reparación consiste en remover uno a uno los ítems de la solución hasta que ésta cumpla con todas las restricciones. Para el caso del problema de la mochila, el cual es el problema Test en el que se evaluará el algoritmo y que se describirá posteriormente en la sección 6.2, el primer ítem en borrar lo determina la máxima relación ganancia/peso q_i la cual se halla mediante la siguiente ecuación:

$$q_i = \max_{j=1}^m \left\{ \frac{c_{ij}}{a_{ij}} \right\}$$

Donde c_{ij} representa la ganancia que aporta la variable de decisión i a la función objetivo j y a_{ij} representa el peso (coeficiente tecnológico) que tiene la variable de decisión i en la restricción j . Se debe tener en cuenta que para el problema de la mochila, el número de objetivos es el mismo número de restricciones, por tanto para otro tipo de problema se debe utilizar otra estrategia para eliminar los ítems.

El ítem con menor q_i es el primero en ser eliminado, así se logra que las variables de decisión que aportan menos ganancia a la función objetivo con respecto a su peso, sean las primeras en ser removidas, hasta poder conseguir un individuo que cumpla con las restricciones mientras que las ganancias de éste se sacrifican lo menos posible.

5.2.2 Dominancia

El concepto de dominancia que se maneja en AGEM es el concepto clásico, donde en un problema de maximización, un individuo i domina a un individuo j , si el individuo i es igual en todos los objetivos al individuo j y es mayor que el individuo j en por lo menos uno de los objetivos.

5.2.3 Factor de Distancia

Estudiando el comportamiento de un algoritmo genético, se pudo observar que estos tienden a hallar todos sus individuos muy cercanos entre si, por tanto la búsqueda puede concentrarse en una región específica del frente de Pareto real, que por lo general es la zona central de éste.

Sin embargo, el interés de toda metodología de algoritmo genético multiobjetivo es hallar una frontera bien distribuida, por tanto se debe lograr que la búsqueda sea expandida más allá que solo la zona central. Es por esto que AGEM hace uso del operador *crowded* propuesto por Deb (2002) para su algoritmo NSGA-II y mencionado en la sección 4.2.6 tanto para hallar la función de evaluación de cada individuo como para seleccionar los individuos que harán parte de la siguiente generación.

5.2.4 Medida de desempeño – Aptitud

AGEM le da prioridad a los individuos que más cantidad de individuos dominen al igual que aquellos que estén más distante del resto de la población, con el fin de hallar una frontera mejor distribuida y más cercana a la real.

Para cumplir con éste propósito, se propone que la medida de desempeño para cada individuo, esté determinada por la siguiente ecuación:

$$f_i = Q_i * (1 + D_i)^2$$

Donde Q_i determina la cantidad de individuos que domina i y D_i representa el factor de distancia de este individuo.

En la primera iteración, tanto la cantidad de individuos que dominan como el operador *crowded* se hallan a partir de los individuos de la población ya reparados, y desde la segunda iteración, donde ya existe un archivo externo llamado Elite, se hallan teniendo en cuenta tanto los individuos de la elite como de la actual generación.

5.2.5 Archivo externo ELITE

En la primera iteración se crea un archivo externo llamado Elite de tamaño constante, donde se guardan las mejores soluciones encontradas en esa iteración. A partir de la segunda iteración este archivo es actualizado con las mejores soluciones que se encuentren. Este es un concepto que se toma de SPEA2 (Zitzler et al., 2001), y la forma como se llena el archivo es una propuesta de esta investigación.

Inicialmente la elite no aceptará soluciones repetidas. Los primeros individuos en hacer parte de ella son los individuos no dominados, luego si el tamaño de la elite no es completado, el archivo se llena con los individuos menos dominados no repetidos, y si finalmente aún el archivo no es completado, entonces se llena con los individuos menos dominados repetidos.

Se propone que los individuos que salen de la elite por ser dominados por algún individuo, pasen a ser parte de la actual generación reemplazando los individuos que salieron para conformar la elite actual.

Los individuos que pertenezcan a la elite al final de la corrida del algoritmo AGEM serán los que finalmente se le muestren al decisor como posibles soluciones.

5.2.6 Selección de Padres

Los padres se seleccionan a partir de los individuos pertenecientes a la elite y a la actual generación, imitando el concepto que propone SPEA (Zitzler y Thiele, 1999) y la selección se hace mediante torneo probabilístico, descrito en la sección 3.3.2.2, donde se propone que compitan un porcentaje del total de posibles padres.

5.2.7 Operadores de Cruzamiento y Mutación

Dado que en los problemas de la vida real se trabaja con un gran número de variables de decisión, entonces se propone hacer uso de la cruce en dos puntos, ya que como se dijo en la sección 3.4.1 el cruzamiento en un punto trae desventajas en éste tipo de problemas.

El operador de mutación varía un poco en comparación al descrito en la sección 3.5, puesto que se propone no hallar una probabilidad de mutación, sino un número de variables para ser mutadas, las cuales serán seleccionadas aleatoriamente. Esto con el fin de minimizar el tiempo computacional.

Además, tomando lo propuesto por algunos investigadores y mencionado en Coello (2004), se decide mutar un mayor número de variables al inicio del proceso y disminuir éste número en la medida que avanza el número de generaciones.

5.2.8 Seleccionar la nueva generación

Tomando el concepto de NSGA-II (Deb et al., 2002) de juntar padres e hijos para seleccionar la nueva generación, se propone que AGEM agrupe la actual generación con la nueva descendencia y a partir de ellos selecciona la nueva población como lo describe Deb (2002).

Para ello organiza tanto la actual generación como los hijos en capas de no dominancia, y crea la nueva generación con los individuos de las primeras capas no dominadas; en el caso de la última capa aceptada, escoge los individuos más distantes con ayuda del operador *crowded*, tal como se describe en la sección 4.2.6.

5.2.9 Diagrama

Con el fin de tener una mayor claridad de cómo trabaja AGEM, se representará su proceso a través de los siguientes diagramas de flujo.

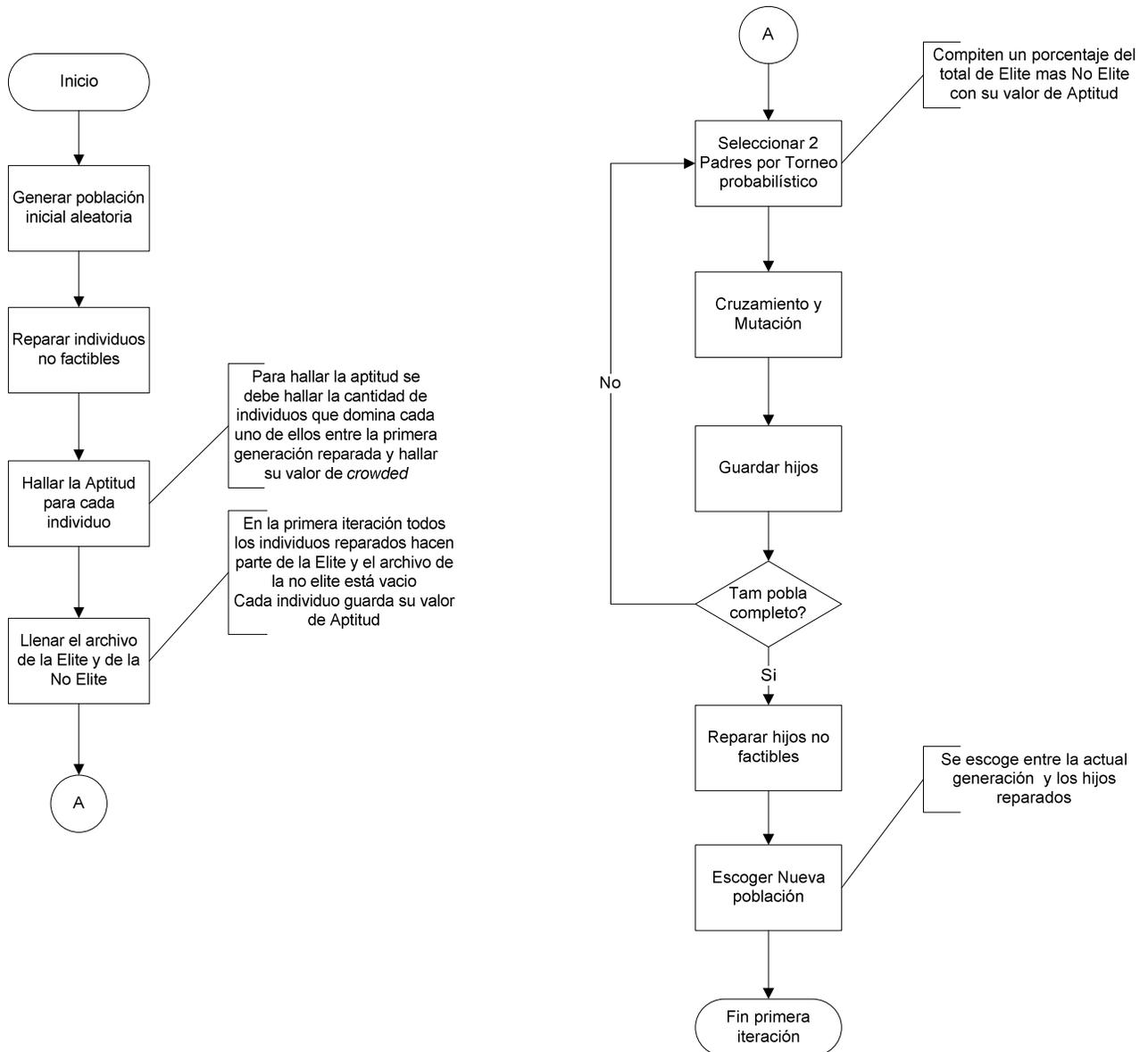


Diagrama 7. Primera iteración de AGEM

A partir de la segunda iteración el proceso varía en el cálculo de la aptitud, puesto que son considerados los individuos que pertenecen a la Elite. El proceso se puede describir en el siguiente diagrama de flujo:

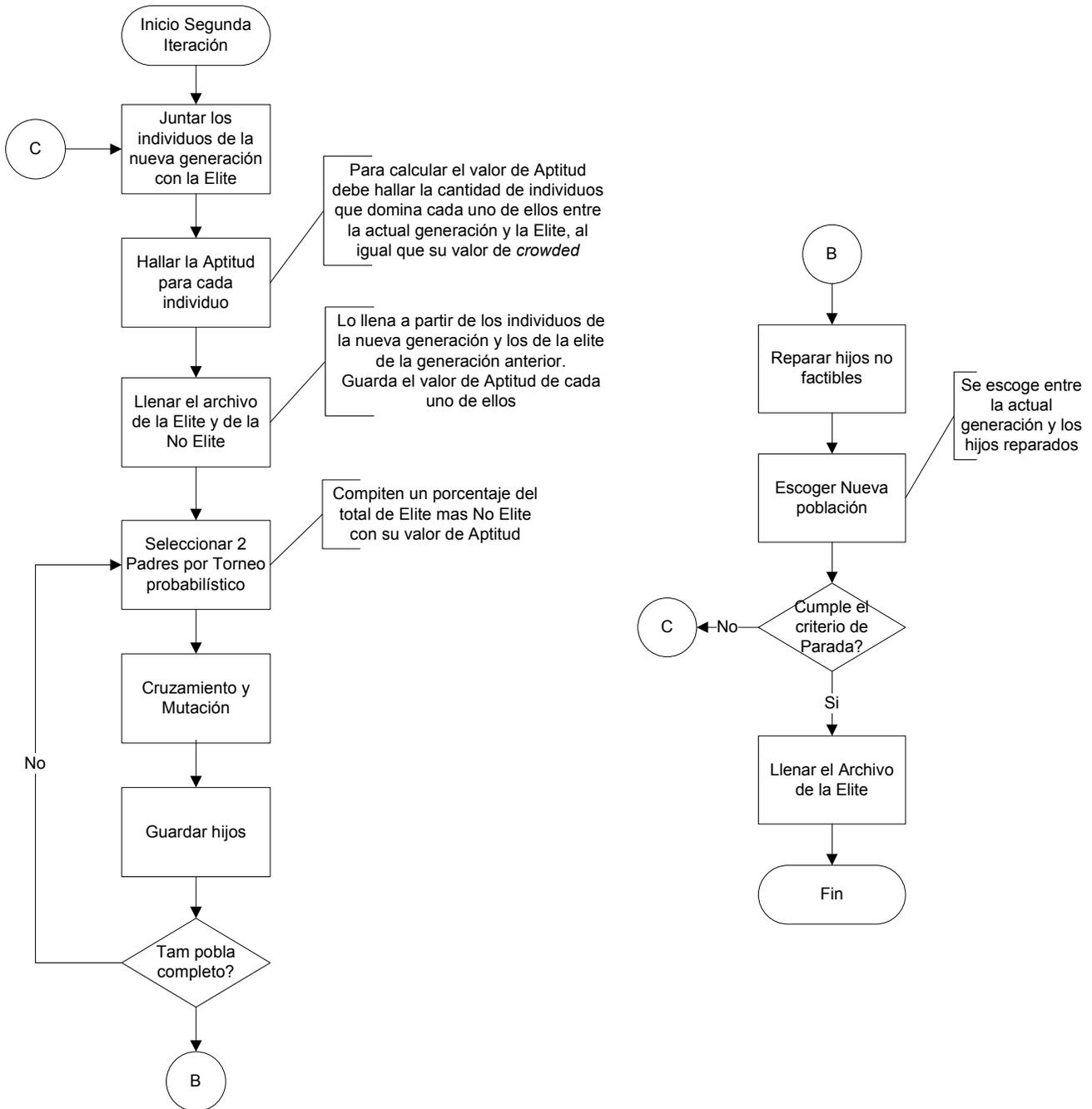


Diagrama 8. Representación en diagrama de flujo del algoritmo AGEM

Los resultados encontrados con esta metodología, aplicados al problema de la mochila muestran mejores características que el SPEA2 y NSGA-II como se mostrará en el apartado 6.5.1

5.3 ALGORITMO GENÉTICO ELITISTA MULTIOBJETIVO QUE INCORPORA LAS PREFERENCIAS DEL DECISOR – AGEM-P

Incorporar las preferencias humanas ha sido un tópico muy importante en la optimización multiobjetivo evolutiva, puesto que la meta final de un problema de optimización no es encontrar toda la frontera de Pareto, sino hallar una alternativa como solución que es la que finalmente el decisor emplearía. Sin embargo, tampoco es útil encontrar una única solución, puesto que el decisor podría sentirse algo incomodo e inseguro con esta, ya que sabe que existen muchas más igualmente eficientes, y no podría tener la opción de conocerlas a menos que vuelva a correr el algoritmo de optimización.

Con la metodología interactiva planteada en esta sección, es posible cumplir con estas dos condiciones, ya que el decisor al ajustar sus preferencias durante el algoritmo finalmente encuentra una fracción pequeña de la frontera de Pareto, en la que muy seguramente estará su solución eficiente preferida.

Además de ello, ésta metodología tiene en cuenta ciertas características que se deben considerar cuando se desea trabajar con preferencias, tales como (Coello, 2000):

- La percepción de una persona siempre estará influenciada por los elementos que se den en un momento determinado y por el ambiente en que se da este momento.
- La función de preferencias o la estructura de valor de una persona no pueden ser expresadas analíticamente, sin embargo, se puede asumir que el decisor tiene un conjunto de creencias adheridas a él.
- Las preferencias del decisor son cambiantes en el tiempo dependiendo su estructura de valor.
- Luego que una persona adquiere experiencia y con ella un aprendizaje, sus aspiraciones y deseos de logro varían.

Para lograr esto, AGEM-P le permite al decisor que antes de definir sus preferencias, pueda percibir lo que realmente está ocurriendo con la solución del problema y que luego de haber adquirido un aprendizaje de éste, él tenga la libertad de definir las veces que lo considere necesario.

Antes de proceder a explicar detalladamente la metodología propuesta, se debe determinar cómo se representará las preferencias del decisor. En la literatura se observó que éstas han sido incluidas de la siguiente forma (Coello, 2000), (Rachmawati y Srinivasan, 2006):

Metodología propuesta

- Mediante pesos dados por el decisor que representan el interés de un objetivo con respecto a los otros. Estos pesos son utilizados en una función escalar con la cual se seleccionan los padres.
- Mediante el nivel mínimo requerido para cada objetivo. Los individuos que no cumplan con el nivel mínimo en un objetivo específico, son penalizados en su función de evaluación.
- Mediante una solución de referencia, es decir, una meta numérica que desea cumplir el decisor. Los padres seleccionados serán los que están más cerca del punto de referencia o meta.
- Hallando la importancia relativa entre objetivos mediante variables lingüísticas y luego convirtiendo estas en pesos o intervalos de pesos que luego son utilizados dentro de una función agregativa para la selección de padres. En el caso de los intervalos, se escoge un peso contenido en él.
- Hallando una función de utilidad que exprese como el decisor combinó los objetivos, a fin de preferir una solución a otra.
- Escogiendo entre dos alternativas, teniendo como base las tasas de compensación entre objetivos. Estas tasas se hallan con los puntos que va obteniendo en el algoritmo.
- Clasificando un conjunto de soluciones en lugar de clasificar los atributos.
- Definiendo previamente metas cualitativas con ayuda de lógica difusa.
- Definiendo dos conjunto de pesos: Lo máximo y mínimo que estaría dispuesto a sacrificar de un objetivo. Con estos pesos se suma linealmente los objetivos ponderados y se trabaja con la dominancia de Pareto.
- Combinando múltiples metas con prioridad entre objetivos, haciendo uso de los operadores *AND* y *OR* (Lógica difusa).
- Mediante un indicador que mide la calidad de la solución, por ejemplo la robustez.

Considerar que el decisor puede expresar sus preferencias a través de pesos, podría ser una situación ideal, puesto que es difícil que una persona pueda expresar su estructura de preferencia en un número explícito de forma absoluta, además, investigadores han demostrado (Coello, 2007) que el uso de pesos a través de funciones agregativas para

resolver problemas multiobjetivo no funciona de manera satisfactoria. Por otro lado, definir funciones de utilidad de manera progresiva para representar los intereses del decisor puede ser algo engorroso, lo mismo que seleccionar o clasificar alternativas de solución en el transcurso del algoritmo, puesto que se requeriría de un gran esfuerzo mental. Ahora bien, el uso de la lógica difusa para representar las preferencias es un área bastante interesante, puesto que asume que el decisor no conoce con exactitud sus preferencias, sino que tiene una idea global de ellas. Sin embargo, en esta investigación no se hará uso de este recurso, pero se considera como trabajo futuro.

Debido a que se busca que el decisor no tenga que hacer un mayor esfuerzo a la hora de definir sus preferencias, puesto que lo debe hacer en varias oportunidades durante la corrida del algoritmo, AGEM-P propone que éste las represente mediante un valor que le proporcione seguridad al momento de definirlo. Este valor podría ser su mínimo esperado.

La autora considera que una persona siempre estará más segura de lo que definitivamente no está dispuesto a aceptar, más que lo que desearía alcanzar, porque generalmente esta última tiende a infinito si está hablando de ganancias o a cero si está hablando de costos.

Por ejemplo, si una persona tuviera que decidir sobre elegir un proyecto entre un conjunto de ellos, teniendo en cuenta las utilidades que estos generan, le sería más fácil decidir cuando realmente conoce que algunos de estos proyectos no cumple con la ganancia mínima exigida por él, por tanto nunca estaría dispuesto a ejecutarlos. Es decir, para el decisor siempre será más fácil decidir acerca de lo que no quiere perder, más que lo que le gustaría ganar.

Sin embargo, estos valores mínimos esperados varían a medida que el decisor aprende del problema, puesto que inicialmente puede que el decisor sea más discreto a la hora de definir lo mínimo que desea obtener, pero cuando se le van presentado posibilidades cada vez más atractivas conforme a sus intereses, él va exigiendo cada vez más en sus valores mínimos, o incluso puede volverse aún más precavido. Por tanto, estos valores mínimos deben ser dinámicos dentro del proceso de búsqueda.

Antes de tomar una decisión sobre los valores mínimos, el algoritmo corre varias iteraciones con AGEM sin tener en cuenta las preferencias, esto con el fin de que el decisor tenga un primer acercamiento con el problema y se sienta más seguro a la hora de definir sus preferencias por primera vez; el número de corridas previas será un porcentaje del total de corridas del algoritmo. Después de definir preferencias, la búsqueda se concentra en zonas más restringidas, por tanto, se considera necesario que el decisor pueda tomar decisiones acerca de la exploración y/o explotación de la búsqueda, para ello

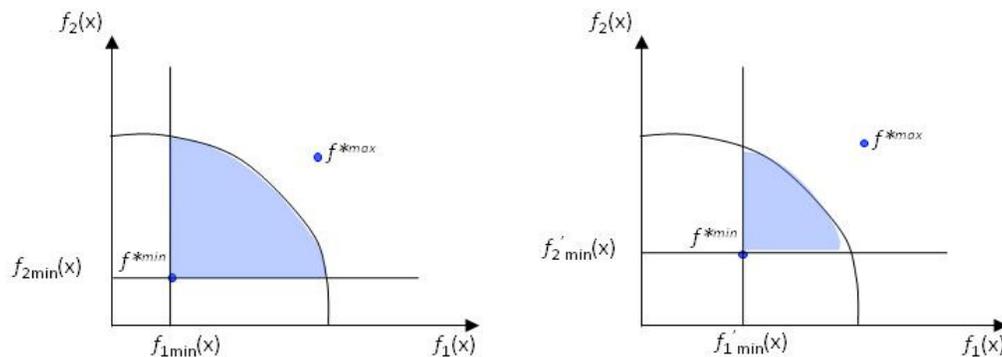
el decisor podrá modificar los parámetros de probabilidad de cruzamiento y el porcentaje de variables a mutar en la medida que el algoritmo avanza.

Después de tomar decisiones acerca de los valores mínimos permitidos y de la explotación y exploración de la búsqueda, el algoritmo trabaja con estos nuevos valores y el decisor puede visualizar mediante gráficas el comportamiento que han tenido las alternativas de la elite y las de la población total; al mismo tiempo que gráficamente puede ver el rango de los valores posibles que toma cada objetivo en toda la población, de esta forma el usuario se cuidaría de no exigirle al problema dar una solución que quizás pueda estar fuera de su alcance. Además de ello, podrá visualizar un promedio de las tasas de intercambio entre sus objetivos, con el fin de que evalúe si le conviene o no aumentar el valor mínimo de uno de sus objetivos.

Con esta información, el decisor podrá tener una idea de cuánto debe disminuir sus exigencias con respecto a un objetivo si deseara aumentar el valor mínimo esperado de otro de ellos; o si no es necesario disminuir sus exigencias sino por el contrario aún es posible aumentar cada una de ellas.

El decisor podrá pausar el algoritmo en el momento que lo considere pertinente para redefinir sus preferencias.

Se esperaría que los mínimos esperados limiten la zona de búsqueda haciendo que ésta sea cada vez menor a medida que avanza el proceso (Ver Gráfica 3) y por tanto la búsqueda se haga mucho más fácil e interesante para el decisor puesto que estaría guiada hacia la fracción de Pareto de su interés, adicional a esto mejoraría la eficiencia en el proceso desde el punto de vista del costo computacional.



Gráfica 3. Limitando el espacio de búsqueda con los mínimos esperados

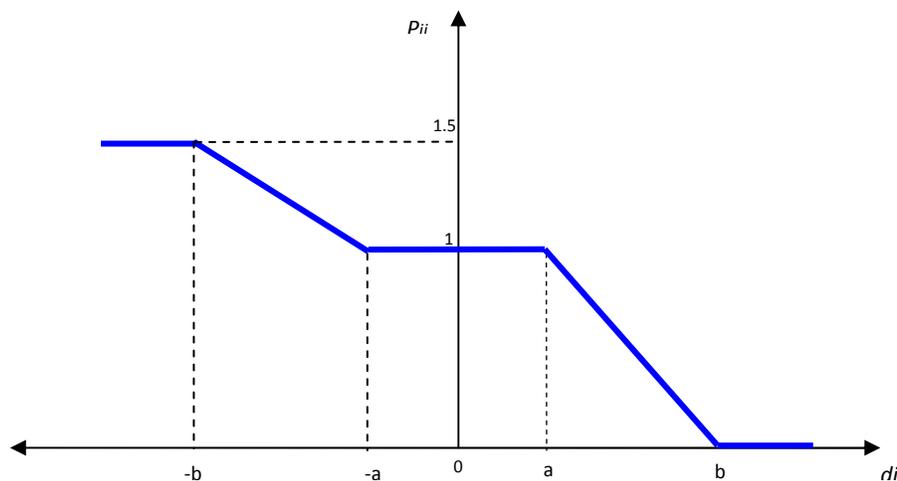
Esta percepción de preferencia será introducida al algoritmo propuesto en la sección 5.2 incorporando un nuevo concepto clave llamado “Factor de Preferencias”, el cual modifica

la dominancia y con ello la función de evaluación y la selección de una nueva generación. También modifica la forma como son aceptados los individuos en la elite.

5.3.1 Factor de Preferencia

Con el fin de evaluar cada alternativa de solución con respecto a las preferencias del decisor, se ha considerado necesario hallar un factor de preferencias FP_i que mida el grado de cumplimiento con los valores mínimos esperados para cada alternativa i .

Debido a que los valores mínimos siguen siendo algo inciertos para el decisor, es decir, no son definidos con una certeza del 100%, entonces se propone hallar el factor de preferencia mediante una función difusa, tal como se representa a continuación:



Gráfica 4. Función del Factor de Preferencia

Donde:

d_i : Es la diferencia entre el mínimo permitido por el decisor y el valor arrojado por la alternativa i en un objetivo determinado. Este valor es negativo cuando el valor del objetivo de la alternativa sobrepasa el mínimo exigido por el decisor.

p_{ij} : Es el factor de preferencia de la alternativa i cuando se evalúa el objetivo j .

a : Es un valor de indiferencia para el decisor donde el objetivo de la alternativa que se está evaluando puede estar a esa distancia del mínimo exigido por el decisor y ser asumido que cumple sus preferencias en un 100%.

b : Es la diferencia máxima permitida por el decisor para decir que una alternativa cumple en algo sus preferencias, es decir, cuando la diferencia sobrepasa este valor el decisor

puede decir con una certeza del 100% que ésta no cumple sus preferencias y por tanto no le interesa.

Con el rango entre a y b el decisor puede expresar su grado de exigencia para el cumplimiento de sus preferencias, siendo más exigente con un rango pequeño y más flexible en la medida que este rango sea más amplio.

El factor de preferencia debe representar que será mucho más importante que un individuo cumpla con sus preferencias en todos los objetivos, de que las cumpla en gran manera en unos cuantos y en otros las viole, es por eso que la función castigará mucho más de lo que premiará, aún cuando la cantidad que supere al valor mínimo esperado sea la misma en la que viole. Por tanto, p_{ij} está en el intervalo $[0.0, 1.5]$ donde toma valores entre $[0.0, 1)$ cuando la alternativa incumple las preferencias dadas por el decisor y entre $[1.0, 1.5]$ cuando es igual o sobrepasa el valor mínimo exigido.

Los valores de a y b al igual que los mínimos esperados, son dados por el decisor después de haber percibido el escenario en el que se encuentra y pueden ser modificados durante la corrida del algoritmo, por tanto podría decidir si ser más exigente o no en las primeras o en las últimas corridas.

Hallado los valores de p_{ij} para cada objetivo j , entonces se define el Factor de preferencia, así:

$$FP_i = \frac{\sum_{j=1}^m p_{ij}}{m}$$

Donde m es el número de objetivos.

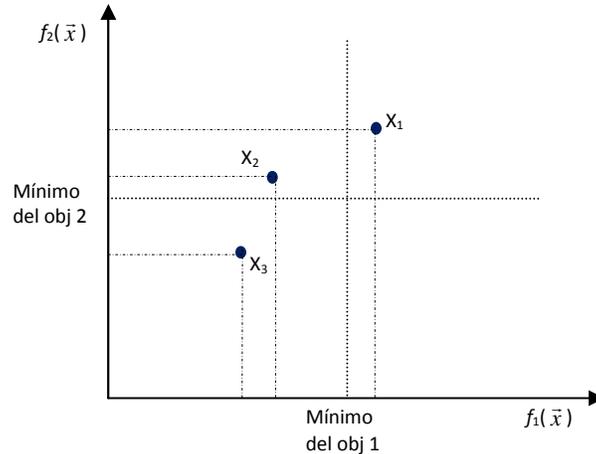
5.3.2 Dominancia por preferencia

El concepto de dominancia para un individuo es probabilístico, donde la alternativa i domina con una probabilidad a la alternativa j , si la alternativa i es igual en todos los objetivos a la alternativa j y es mayor que la alternativa j en por lo menos uno de los objetivos, siendo la probabilidad el factor de preferencia de la alternativa i (FP_i).

Así, un individuo i con $FP_i \geq 1$ y que domine a otro por sus valores de objetivos, entonces siempre será dominador, a su vez, un individuo con $FP_i < 1$ que domine a otro por sus valores de objetivos, tendrá una probabilidad de fallar en ser efectivamente dominante,

de igual forma un individuo con $FP_i = 0$ nunca será dominante a pesar de sus valores en los objetivos.

En el caso de la Gráfica 5, donde se tienen dos objetivos, y asumiendo que $FP_1 = 1.2$, $FP_2 = 0.85$, y $FP_3 = 0.4$, la alternativa 1 siempre dominará a las alternativas 2 y 3, y la alternativa 2 tiene una probabilidad de 0.85 de dominar a la alternativa 3.



Gráfica 5. Dominancia teniendo en cuenta las preferencias del decisor

No se acepta de una forma determinística la dominancia de la alternativa 2, puesto que esta última no cumple con el valor mínimo esperado del decisor para el objetivo 1 y dejarle ser dominante a esta alternativa es permitirle que continúe reproduciéndose en las siguientes generaciones y que lleve la población a zonas que no son de interés para el decisor.

De esta forma se les daría más importancia a aquellos individuos que realmente cumplen o están muy cerca de cumplir con las preferencias del decisor.

5.3.3 Función de Evaluación – Aptitud

En la sección 5.2.4 la función de evaluación para AGEM depende del factor de distancia de cada individuo y del número de individuos que cada uno domina.

En el caso de incorporar las preferencias del decisor, no es necesario que los individuos estén lo más disperso posible ya que se requiere que la búsqueda sea en zonas específicas, por tanto no se empleará el factor de distancia. Sin embargo, el número de individuos que domina continúa siendo parte de este cálculo, teniendo en cuenta que ahora el concepto de dominancia ha cambiado.

Dado que las preferencias dentro de la dominancia son asumidas sólo como una probabilidad, en el momento de definir el valor de actitud para cada individuo es necesario volverlas a tener en cuenta, en este caso el factor de preferencia hace las veces de castigo o premio a la función de evaluación dependiendo si viola o no las preferencias del decisor.

Un individuo entre más viole las preferencias tendrá una menor probabilidad de ser reproducido que uno que menos las viole. Y en caso contrario, un individuo entre mas cumple las preferencias tendrá una mayor probabilidad de ser seleccionado como padre.

Ahora, la función de evaluación está dada por la siguiente ecuación:

$$f_i = Q_i * FP_i^2$$

5.3.4 Archivo Externo ELITE

Se debe recordar que en la Elite serán guardados los individuos que finalmente harán parte de la solución presentada al decisor, por tanto es importante que en este archivo permanezcan aquellos individuos interesantes para él.

Como este archivo tiene un tamaño fijo, es ahora completado por los siguientes individuos y en el mismo orden:

- Individuos no dominados, no repetidos y que además cumplen completamente con los valores mínimos esperados.
- Individuos menos dominados, no repetidos y que además cumplen completamente con los valores mínimos esperados.
- Individuos menos dominados, no repetidos, sin importar que no cumplan totalmente con los valores mínimos esperados.
- Individuos menos dominados, sin importar que sean repetidos o que no cumplan totalmente con los valores mínimos esperados.

En el caso que el primer grupo de individuos sea mucho mayor al tamaño máximo de la Elite, entonces se deberá seleccionar de ellos los individuos que tenga mayor valor de FP_i .

En la última iteración sólo se presentan los que de la elite cumplen totalmente con las preferencias del decisor.

5.3.5 Seleccionar la nueva generación

La nueva generación se selecciona por capa de no dominancia, tal como lo plantea AGEM, considerando que el concepto de dominancia varía.

5.4 CONCLUSIONES

Debido a que para el decisor es difícil definir sus preferencias de manera determinística, AGEM-P le da la posibilidad de reevaluarlas cada vez que lo considere necesario, no sin antes permitirle visualizar el panorama en el que se encuentra.

AGEM-P, utiliza como base el algoritmo propuesto AGEM, sin embargo debido al manejo que se les da a las preferencias, esta misma metodología podría ser utilizada en otros algoritmos que trabajen con el concepto de dominancia, tal como el SPEA2 y el NSGAI.

Dado que es necesario evaluar las metodologías propuestas, en el siguiente capítulo se escoge un problema Test combinatorial, el cual se aplicará a los dos algoritmos de referencia y a los propuestos en esta investigación con el fin de evaluar su aporte al estado del arte.

6 RESULTADOS DE LA SIMULACIÓN

6.1 INTRODUCCIÓN

Con el fin de validar las metodologías propuestas en el capítulo 5, se escogerá un problema Test combinatorial muy conocido en la literatura llamado Problema Multiobjetivo de la Mochila y este se resolverá con cada metodología para 2 y 3 objetivos variando el número de artículos posibles entre 400, 600 y 800 artículos.

Para determinar si AGEM obtiene o no resultados más cercanos de la frontera de Pareto real y con una distribución uniforme a lo largo de la frontera encontrada, se calculará el valor de las métricas *Generational Distance* – GD y *Spacing* – S (Coello et al., 2002), y se compararán con los valores que tomen estas para los algoritmos NSGAII y SPEA2, de esta forma se concluirá si el algoritmo propuesto puede obtener mejores soluciones que los algoritmos de referencia.

En el caso de AGEM-P, se validará resolviendo el mismo problema pero para dos objetivos, con el fin de mostrar gráficamente en un plano cartesiano los resultados y demostrar si realmente es necesario que el decisor incorpore sus preferencias durante la corrida del algoritmo y no antes o después de ella.

Inicialmente se describirá el problema test combinatorial (sección 6.2) y las métricas que se emplearan para medir la eficiencia de la nueva metodología – AGEM (sección 6.3). Luego se mencionará la configuración utilizada para cada uno de los algoritmos (sección 6.4) y finalmente se mostrarán los resultados tanto para AGEM como para AGEM-P (sección 6.5), definiendo la utilidad de cada uno de ellos.

6.2 PROBLEMA TEST

Con el fin de evaluar la metodología propuesta se ha escogido un problema de optimización clásico llamado Problema Multiobjetivo de la mochila (*Multiobjective 0/1 Knapsack Problem*) el cual es muy conocido en la literatura y calificado por Zitzler y Thiele (1999) como un problema general, comprensible, fácil de formular y que representa un tipo de problema en el mundo real.

Este problema básicamente consiste en seleccionar un número de artículos que tienen asociados a ellos una ganancia y un tamaño. Estos artículos serán introducidos en una mochila que tiene una capacidad limitada y se busca maximizar la ganancia total obtenida al escoger los artículos.

Inicialmente este problema fue diseñado para un solo objetivo y luego fue ajustado para múltiples objetivos haciendo referencia a varias mochilas.

La formulación de éste problema multiobjetivo es definida de la siguiente manera (Zitzler y Thiele, 1999):

Sea n la cantidad de artículos y m el número de mochilas.

c_{ij} = Ganancia del artículo i metido en la mochila j

a_{ij} = Tamaño del artículo i metido en la mochila j

b_j = Capacidad de la mochila j

Se debe hallar el vector solución $x = (x_1, x_2, x_3, \dots, x_n) \in \{0,1\}^n$ el cual toma valores de 1 cuando el artículo es seleccionado y de 0 cuando no lo es, tal que:

$$\text{Max} f(x) = (f_1(x), f_2(x), \dots, f_m(x))$$

Donde:

$$f_j(x) = \sum_{i=1}^n c_{ij} x_i$$

Sujeto a:

$$\sum_{i=1}^n a_{ij} x_i \leq b_j \quad \forall j \in \{1, 2, \dots, m\}$$

Este problema se resolverá para dos y tres mochilas, con el fin de evaluar la metodología para dos y tres objetivos, variando en cada uno de ellos la cantidad de artículos entre 400, 600 y 800. Los datos que se utilizaron, fueron calculados de la misma forma que se propone en Zitzler y Thiele (1999) con la diferencia que se exigirá un poco más, ya que se aumentará el rango de valores para las ganancias y el tamaño de cada artículo y se disminuirá la capacidad total para cada mochila. Las ganancias y el tamaño de cada artículo son dadas por números aleatorios entre 10 y 150 y la capacidad total para cada mochila será el 40% de la suma de los tamaños de los artículos en cada mochila.

Los datos se presentan en el ANEXO C.

6.3 MÉTRICAS DE EVALUACIÓN

Dos parámetros fundamentales que se tienen que tener en cuenta a la hora de evaluar un algoritmo genético multiobjetivo son la cercanía de sus resultados a la frontera de Pareto real y la distribución de éstos a través de ella. En la actualidad existen muchas métricas que permiten comparar varios algoritmos con relación a estos dos criterios, las cuales se mencionan en Coello *et al.* (2002). En esta investigación se hará uso de dos de ellas.

6.3.1 Distancia Generacional (*Generational Distance – GD*)

GD representa en promedio que tan lejos puede estar una frontera no dominada de la frontera real de Pareto. Esta métrica se obtiene a partir de la siguiente ecuación:

$$GD = \frac{\left(\sum_{i=1}^n d_i^p \right)^{1/p}}{n}$$

Donde n es el número de individuos que se hallaron en la frontera, $p = 2$, y d_i es la distancia euclidiana en el espacio de los objetivos entre un individuo de la frontera hallada y el más cercano de la frontera real.

Note que cuando d_i es cero, significa que coincide el individuo hallado con el de la frontera real, entonces, mientras este índice tenga un valor más cercano a cero será mejor, puesto que indicaría que la frontera está más cerca a la real.

6.3.2 Distribución (*Spacing – S*)

Ésta métrica representa que tan uniformemente están distribuidos los individuos a lo largo de la frontera encontrada. Se define como:

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

Equivalente a la desviación estándar muestral de los d_i . Donde n es el número de vectores en la frontera no dominada, d_i es la distancia del individuo i al individuo j más cercano, el cual se define para dos objetivos como:

$$d_i = \min_j \left(\left| f_1^i(\vec{x}) - f_1^j(\vec{x}) \right| + \left| f_2^i(\vec{x}) - f_2^j(\vec{x}) \right| \right) \quad i, j = 1, \dots, n$$

Y \bar{d} es la media de todos los d_i .

En el caso de un problema con m objetivos, la autora redefinirá el valor de d_i como sigue:

$$d_i = \min_j \left(|f_1^i(\bar{x}) - f_1^j(\bar{x})| + |f_2^i(\bar{x}) - f_2^j(\bar{x})| + \dots + |f_m^i(\bar{x}) - f_m^j(\bar{x})| \right)$$

Cuando ésta métrica toma el valor de cero, significa que todos los individuos de la frontera hallada son distribuidos equidistantemente.

Estas dos métricas serán utilizadas para evaluar si el algoritmo propuesto AGEM mejora en los dos criterios mencionados inicialmente con respecto a los algoritmos NSGAI y SPEA2.

6.4 CONFIGURACIÓN DEL ALGORITMO

Para todos los problemas se han corrido 500 iteraciones con los siguientes parámetros:

PARÁMETRO	NSGAI	SPEA2	AGEM	AGEM-P
Tamaño de la Población	250	250	250	250
Tamaño de la Elite	-	250	250	80
Puntos de Cruzamiento	2	2	2	2
Prob de Cruzamiento	0.9	0.9	0.9	Inicialmente 0.9 luego lo define el usuario
Prob de Mutación	0.006	0.006	Se disminuye progresivamente durante el proceso	Inicialmente 2% de las variables, luego lo define el usuario
K Selección Padres	0.75	0.75	0.75	0.75
Tipo de Torneo	Binario	Binario	Compite 10% de la Población	Compite 10% de la Población

Tabla 1. Parámetros de los Algoritmos

En AGEM se propone una mutación variada, puesto que luego de estudiar gráficamente el comportamiento de la población durante varias corridas se notó que se obtiene mejores resultados cuando se permite una mayor exploración al principio de las iteraciones y se disminuye ésta al finalizar. La autora plantea que esto se debe a que a medida que la

solución hallada se acerca a la frontera real de Pareto, la zona de búsqueda limitada por las restricciones será menor, por tanto si se tiene una mayor exploración será más difícil encontrar buenas soluciones dentro de la zona factible, pero una mayor explotación hará que se encuentren soluciones cercanas a las ya halladas y mucho mejores.

El número de iteraciones antes de definir preferencias en AGEM-P, será del 10% del total de iteraciones. Este trabaja con una Elite constante de 80 individuos durante toda la corrida y sólo al final de ésta, la elite solo recibirá los individuos que son no dominados y que además cumplen totalmente con las preferencias finales del decisor.

Con el fin de comparar los algoritmos, fue necesario programar NSGAI y SPEA2. Esto se hizo siguiendo la descripción hecha en Deb *et al.* (2002) para NSGAI y en Zitzler *et al.* (2001) para SPEA2. Tanto estos como AGEM y AGEM-P fueron programados en C# con Microsoft Visual Studio 2005.

6.5 RESULTADOS

Dado que se proponen dos algoritmos, los resultados serán mostrados en dos secciones. Una de ellas mostrará los resultados de AGEM con respecto a las métricas de GD y S para cada uno de las combinaciones Objetivos – Variables en comparación con NSGAI y SPEA2, y en otra sección, se comparará gráficamente los resultados de AGEM y los de AGEM-P, verificando si involucrar las preferencias de una forma interactiva realmente logra mejorar las expectativas de un decisor.

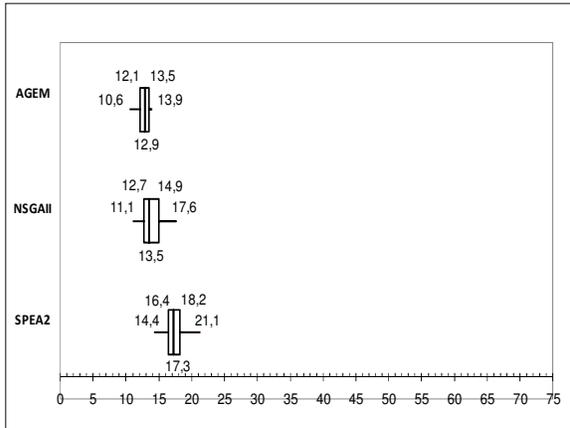
6.5.1 Resultados experimentales de AGEM

Se utilizaron los mismos datos para cada algoritmo según el problema; la población inicial fue totalmente aleatoria, por tanto era necesario para cada problema hacer varias corridas por algoritmo y así poder identificar el comportamiento de las métricas en cada uno de ellos. Para cada corrida se hallaron las métricas GD y S , lo cual hizo necesario calcular la frontera real con ayuda del software GAMS que resuelve los problemas con algoritmos clásicos de optimización. Las métricas son analizadas mediante diagramas de cajas y bigotes en donde se visualizan los valores mínimos y máximos, los cuartiles Q1 y Q3, y el valor de la mediana.

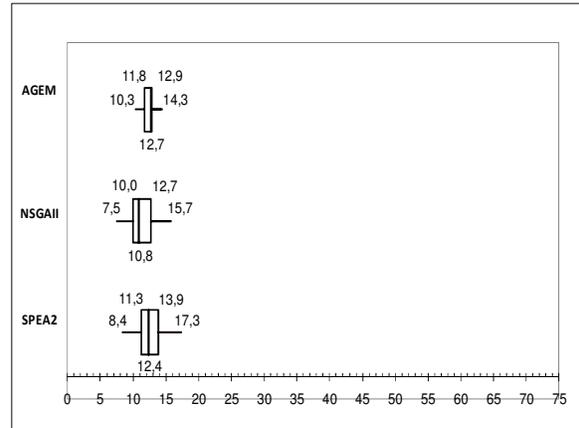
Luego de realizar 15 corridas por cada algoritmo, para resolver un problema determinado, se obtuvieron los siguientes resultados:

Para el problema de la mochila con 2 objetivos y 400 variables de decisión (Ver Gráfica 6), el valor de la mediana es menor para AGEM que para NSGAI, por tanto el 50% de las

veces la frontera de Pareto hallada por AGEM fue más cercana de la frontera real que el 50% de la veces obtenida por NSGAI. Además, el 100% de las veces fue más cercana que las halladas por SPEA2; sin embargo, las fronteras halladas con NSGAI y SPEA2, en la mayoría de los casos, fueron mejor distribuidas que las obtenidas por AGEM (Ver Gráfica 7).



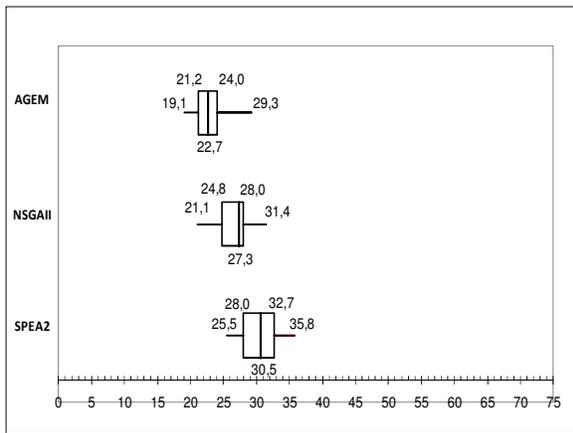
Gráfica 6. Métrica GD para 2 objetivos 400 variables



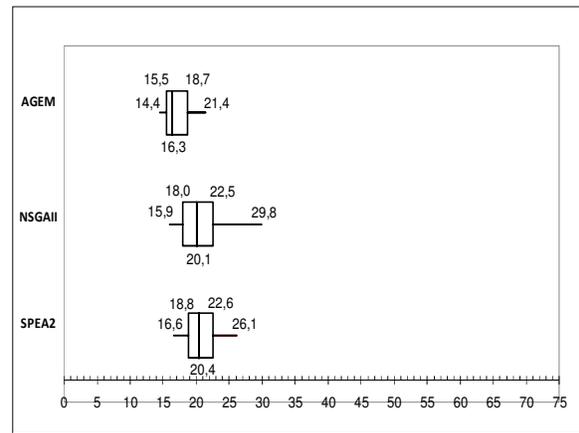
Gráfica 7. Métrica S para 2 objetivos 400 variables

Cuando se aumenta el número de variables a 600, AGEM mejora la distribución en sus fronteras en comparación a los otros dos algoritmos (Ver Gráfica 9) y continúa obteniendo fronteras más cercanas a la real que NSGAI y SPEA2 (Ver Gráfica 8). Por ejemplo, AGEM encontró el 75% de la veces fronteras más cercanas que SPEA2 y a su vez, el 50% mejor distribuidas.

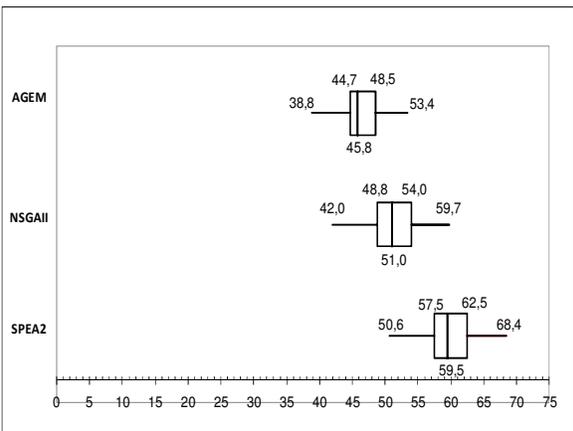
En la Gráfica 10, se puede observar que cuando el número de variables de decisión asciende a 800, AGEM continúa encontrando fronteras más cercanas a la real, se puede notar que el valor de la mediana de GD para AGEM es mucho menor que para NSGAI y SPEA2. Igualmente ocurre con la distribución uniforme de las fronteras halladas (Gráfica 11), pues los valores obtenidos por AGEM en la métrica S, el 75% de las veces son menores que los obtenidos por NSGAI y SPEA2.



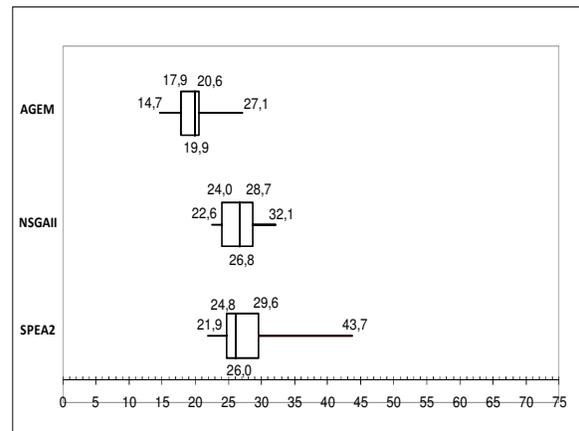
Gráfica 8. Métrica GD para 2 objetivos 600 variables



Gráfica 9. Métrica S para 2 objetivos 600 variables



Gráfica 10. Métrica GD para 2 objetivos 800 variables



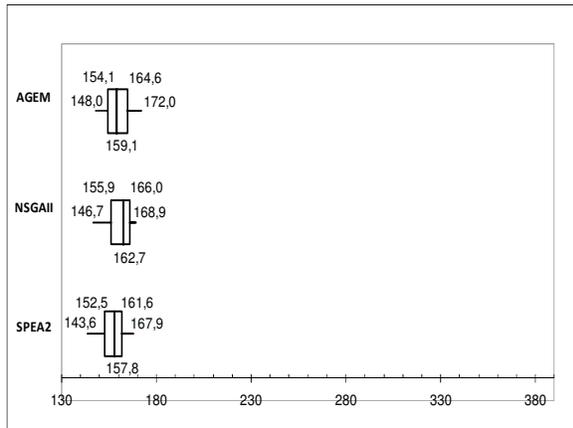
Gráfica 11. Métrica S para 2 objetivos 800 variables

En la medida que se aumenta el número de las variables, las fronteras halladas en los tres algoritmos se alejan de la frontera real, esto es porque se mantiene el número de las corridas en 500 aún cuando se tiene un espacio de búsqueda mucho mayor. En el caso de la distribución de las fronteras (Métrica S) se mantiene un poco más, siendo las de AGEM las que mantienen una mejor distribución uniforme entre sus individuos.

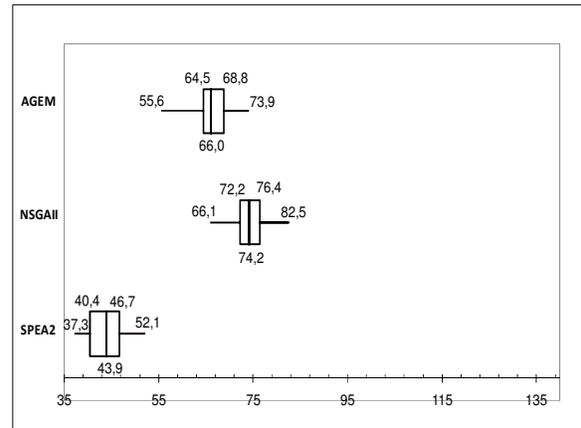
En el caso del problema de la mochila con 3 objetivos y 400, 600 y 800 variables, los algoritmos se comportan de forma similar con respecto a la cercanía a la frontera real, (Ver Gráfica 12, Gráfica 14, Gráfica 16) no obstante, en todos los casos, la distribución a lo largo de la frontera es mucho mejor con SPEA2 que con AGEM y NSGAI (Ver Gráfica 13, Gráfica 15 y Gráfica 17).

Resultados de la simulación

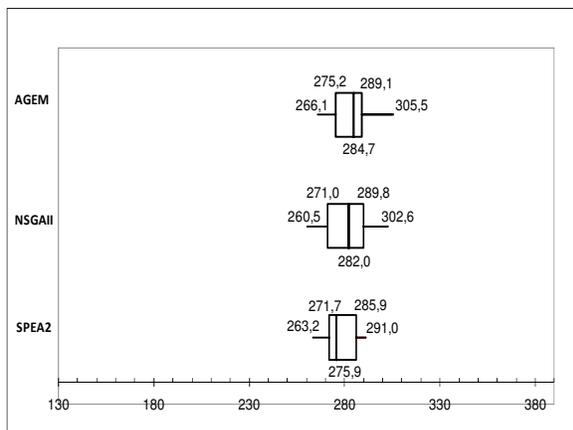
Sin embargo, a pesar de que AGEM utilice la misma técnica que NSGAI para distribuir los individuos a lo largo de la frontera, el primero tiene un mejor indicador que este último en todos los casos.



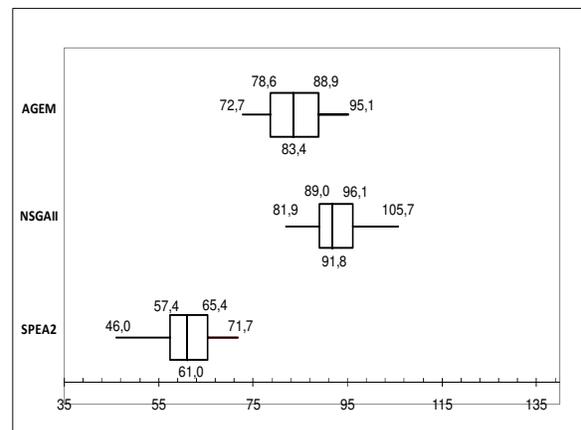
Gráfica 12. Métrica GD para 3 objetivos 400 variables



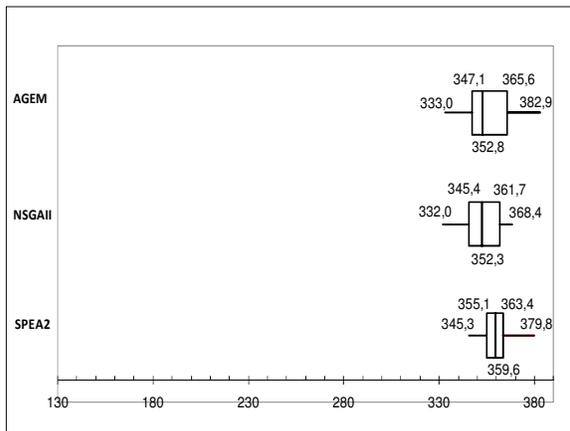
Gráfica 13. Métrica S para 3 objetivos 400 variables



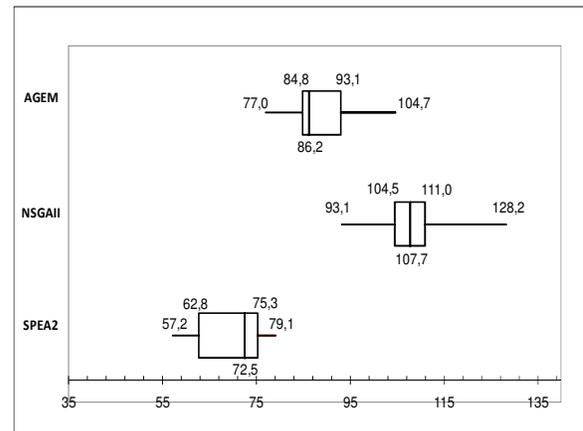
Gráfica 14. Métrica GD para 3 objetivos 600 variables



Gráfica 15. Métrica S para 3 objetivos 600 variables



Gráfica 16. Métrica GD para 3 objetivos 800 variables



Gráfica 17. Métrica S para 3 objetivos 800 variables

En resumen, AGEM, en la mayoría de las corridas, obtuvo fronteras más cercanas a la real cuando se resolvió el problema de la mochila con 2 objetivos para 400, 600 y 800 variables, y en el caso del problema de la mochila con 3 objetivos, no se pudo identificar con claridad cuál de los tres algoritmos obtiene la frontera más cercana a la real.

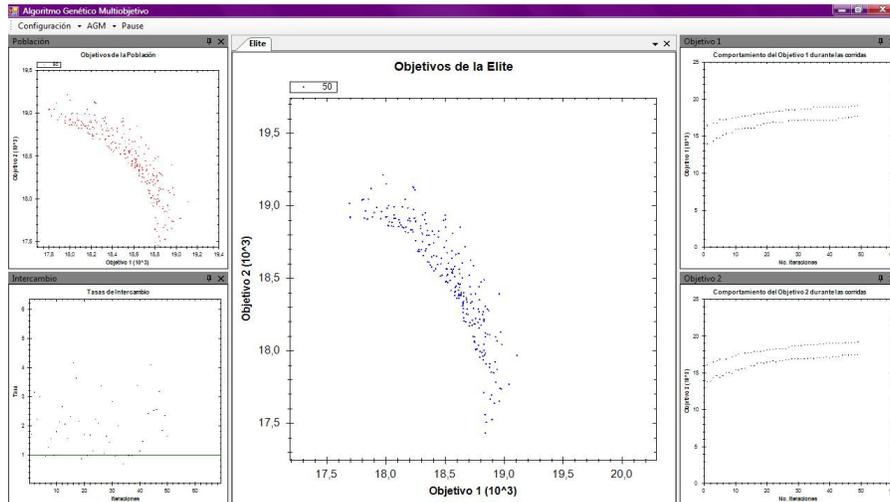
Con respecto a la distribución de las fronteras halladas, AGEM, en la mayoría de las iteraciones, obtiene mejor distribución cuando se resuelve el problema de la mochila para 2 objetivos con 600 y 800 variables, no ocurriendo lo mismo para 400 variables. Y en el caso de 3 objetivos, SPEA2 siempre obtiene la mejor distribución, pero se debe resaltar que AGEM presenta una mejor distribución que NSGAI aún cuando utiliza la misma técnica de distribución.

Por tanto, se podría afirmar que AGEM al guardar en cada iteración los mejores individuos para la selección de los padres, y al no olvidar los buenos individuos hallados en la población anterior para continuar con su búsqueda, arroja mejores resultados que si se considerarían los dos conceptos por separados.

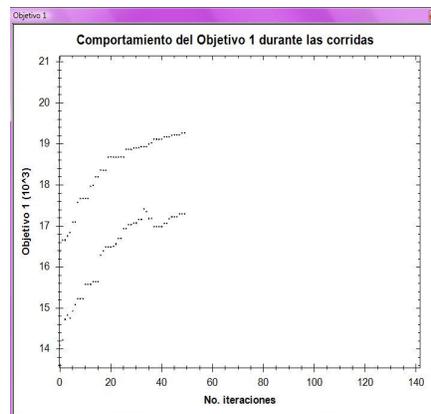
6.5.2 Resultados experimentales de AGEM-P

Con el fin de validar la propuesta de involucrar las preferencias de una manera interactiva en un algoritmo genético multiobjetivo (AGEM), se hizo un aplicativo donde el decisor puede resolver el problema de la mochila con 2 objetivos, tanto para 400 como para 600 y 800 variables (Ver ANEXO B). Se escogió un problema de 2 objetivos, debido a que es posible visualizar la frontera de Pareto en un plano XY y por tanto se puede percibir gráficamente si los valores obtenidos en los objetivos son superiores a los obtenidos por la metodología sin usar preferencias o están en zonas que ésta última no tiene en cuenta.

Durante todo el proceso, el algoritmo le permite visualizar al decisor el comportamiento de la población elite (en el centro), de toda la población (Superior Izquierda) y el rango en el que se encuentran los valores de sus objetivos (Gráficas a la derecha). Así como el comportamiento de las tasas promedio de intercambios entre sus objetivos (Inferior Izquierda). Esto se puede observar en la Gráfica 18 y Gráfica 19.

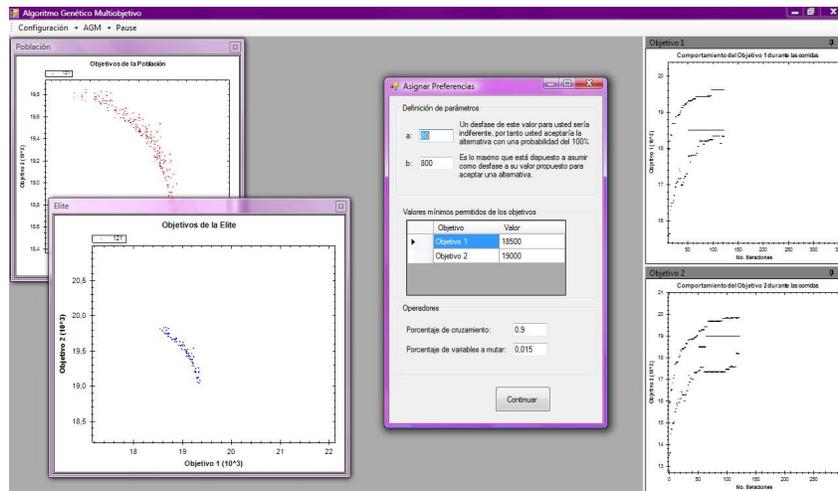


Gráfica 18. Información que observa el decisor antes de definir sus preferencias



Gráfica 19. Rango del Objetivo 1

Con esta información el decisor tendrá una idea para definir sus preferencias por primera vez, y luego en la medida que observa el proceso, notará el comportamiento de las soluciones queriendo intervenir en él en cualquier momento, por tanto se le da la opción para que pause el algoritmo cuando lo desee de tal forma que pueda redefinir sus preferencias. En el momento de pausar el algoritmo, también muestra un mensaje donde de acuerdo a la tasa promedio de intercambio obtenida en la iteración actual, se le indica si es conveniente o no aumentar el valor mínimo de un objetivo determinado.

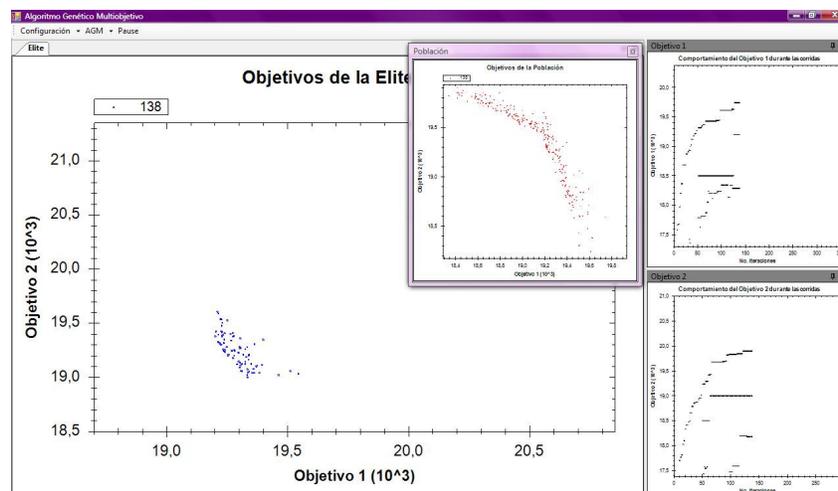


Gráfica 20. Algoritmo pausado para redefinir preferencias

De esta forma el decisor podrá evitar tempranamente que la búsqueda se extienda hacia zonas que no le interesan.

Con la información que percibe el decisor, puede ver el comportamiento que van teniendo los valores en cada una de las funciones objetivo, así como las tasas promedio de intercambio entre ellos.

Luego de estudiar el comportamiento de las soluciones después de que el decisor defina sus preferencias, se observa como las soluciones de la elite (en azul) se mueven inicialmente hacia la zona limitada por las preferencias, y posterior a esto continúa su búsqueda hacia la frontera de Pareto (Ver Gráfica 21).

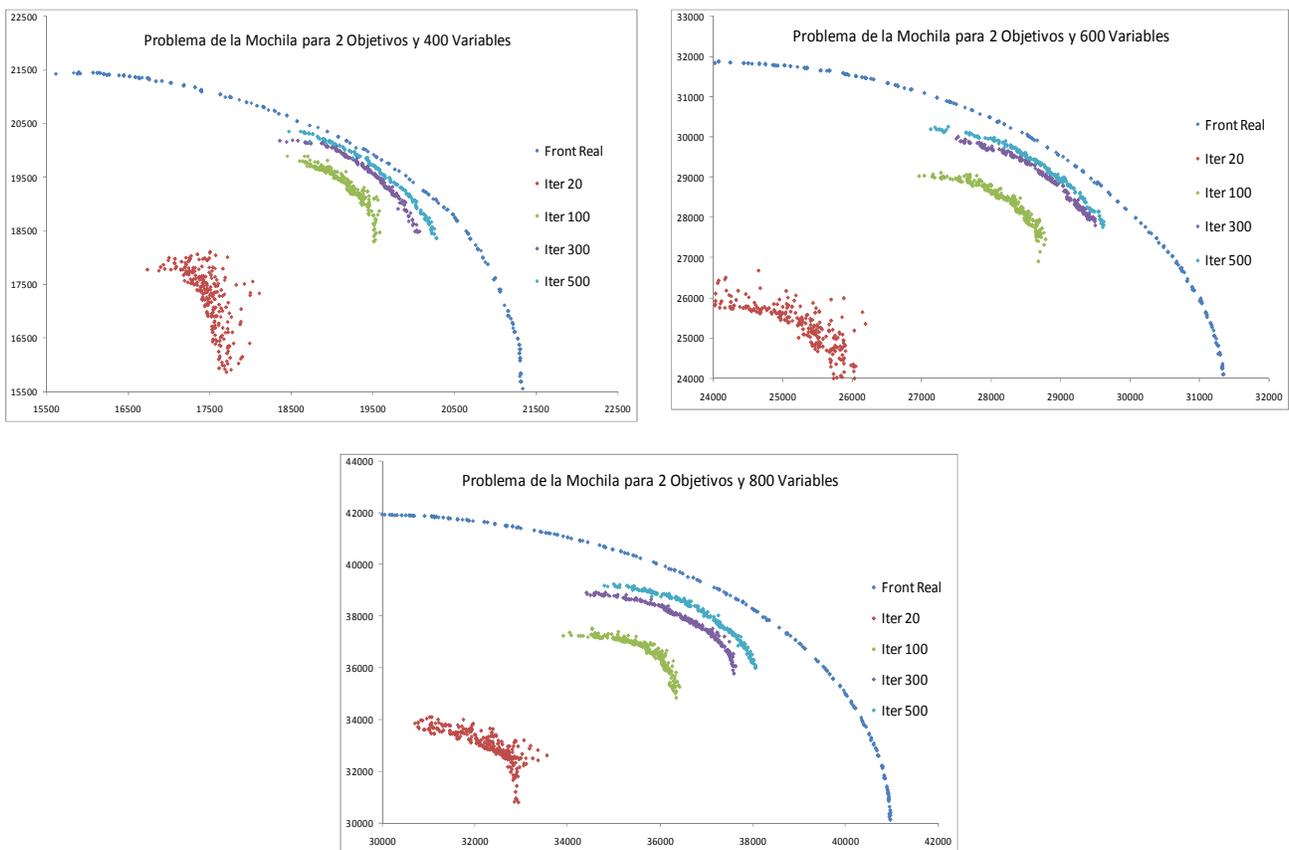


Gráfica 21. Movimiento de la Elite hacia la nueva zona de búsqueda

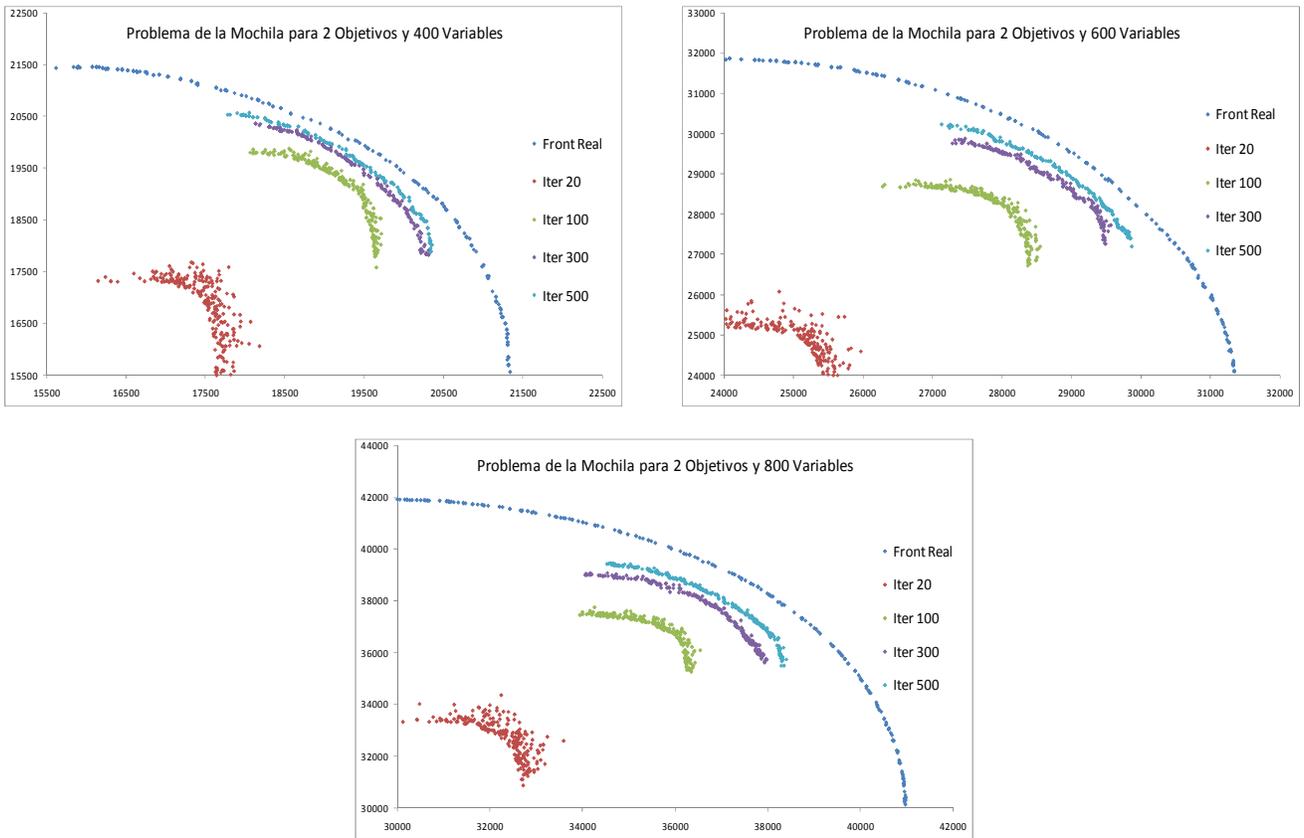
Resultados de la simulación

Sin embargo, en la medida que el proceso se acerca a su final, se tienen zonas cada vez más restringidas, por lo cual, el algoritmo no podría avanzar eficientemente con los mismos parámetros de cruzamiento y mutación con el que inicio el proceso. Pero, cuando disminuye la mutación (exploración) y/o aumenta el porcentaje de cruzamiento (explotación), puede permitirle avanzar libremente al algoritmo.

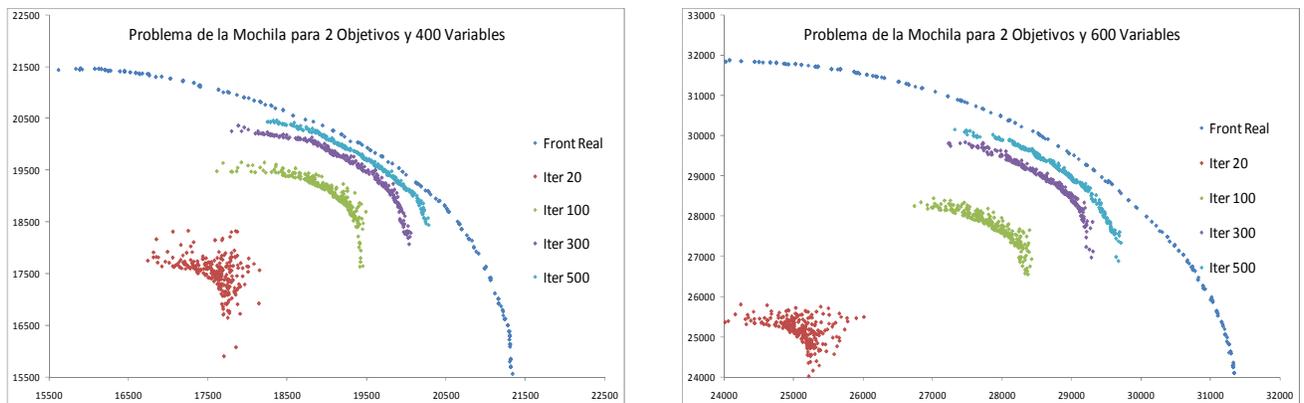
Durante varias pruebas se comprobó que los algoritmos genéticos multiobjetivo como el NSGAI, SPEA2 y aún AGEM en problemas como la mochila con dos objetivos siempre tienden a buscar la zona central de la frontera de Pareto tal como se puede visualizar en las siguientes gráficas.

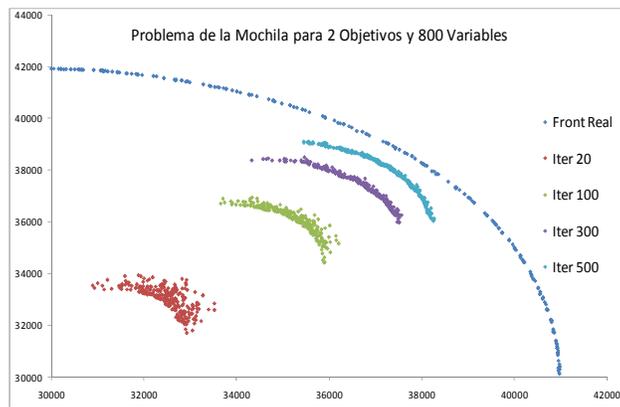


Gráfica 22. Proceso de búsqueda de NSGAI para el problema de la Mochila con 400, 600 y 800 variables y 2 Objetivos



Gráfica 23. Proceso de búsqueda de SPEA2 para el problema de la Mochila con 400, 600 y 800 variables y 2 Objetivos

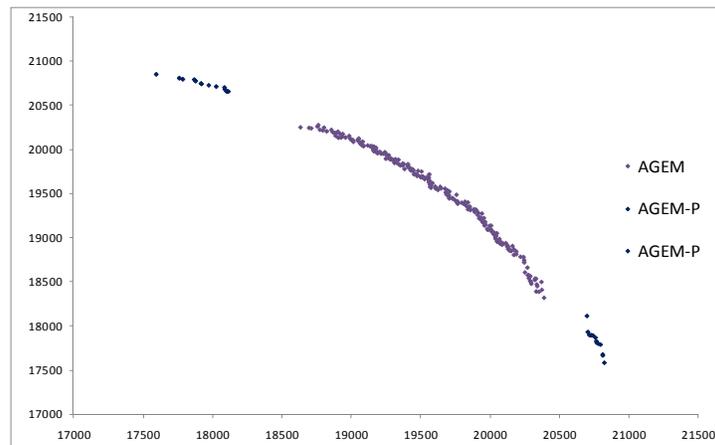




Gráfica 24. Proceso de búsqueda de AGEM para el problema de la Mochila con 400, 600 y 800 variables y 2 Objetivos

Por tanto, si el decisor está interesado en uno de los extremos de la frontera, una de estas metodologías no podría ofrecérselo. Sin embargo, en la Gráfica 25 se puede observar que al incorporar las preferencias del decisor de forma interactiva en AGEM, es posible obtener soluciones que cumplan con este interés.

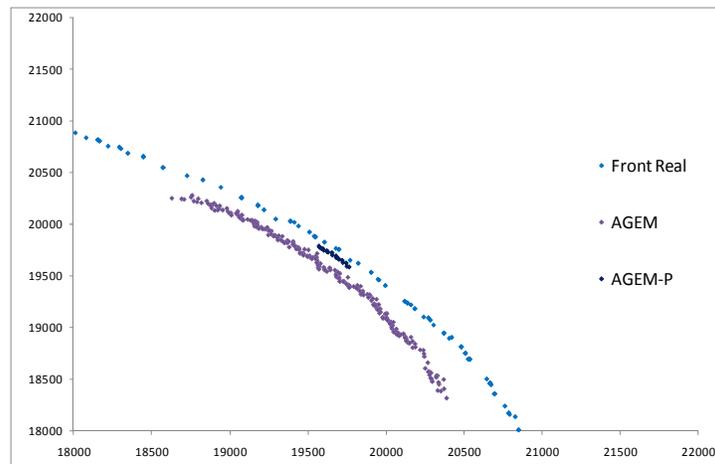
En esta gráfica se observa la frontera de Pareto obtenida por AGEM (en morado) y las fracciones de la frontera obtenidas cuando se corre AGEM-P (en azul) dándole mayor importancia a un objetivo y luego dándole mayor importancia al otro.



Gráfica 25. Comparación entre AGEM y AGEM-P para extremos en el problema de la Mochila con 2 Objetivos y 400 Variables

Este algoritmo interactivo, además de considerar las zonas donde el algoritmo no logra buscar por sí solo, al restringir su zona de búsqueda es capaz de superar las soluciones halladas por una metodología que no consideran las preferencias, es decir, se acerca aún

más a la frontera real de Pareto en el mismo número de iteraciones y bajo los mismos parámetros considerados en un algoritmo que no involucren las preferencias de una forma interactiva, exceptuando el porcentaje de cruzamiento y mutación que podrán ser modificados por el decisor. Esto se debe a que parte del costo computacional que no desperdicia en encontrar un gran número de soluciones y al interior de un amplio espacio de búsqueda, que al final definitivamente descartará, lo usa en encontrar soluciones más eficientes de la porción de la frontera de Pareto que verdaderamente le interesa. Esto se puede observar en la Gráfica 26.



Gráfica 26. Comparación entre AGEM y AGEM-P para zonas en común en el problema de la mochila con 2 Objetivos y 400 Variables

Al final del proceso de búsqueda de la frontera de Pareto, comienza un proceso de análisis multiatributo en el que necesariamente el decisor tendrá que incorporar sus preferencias para encontrar la única alternativa que definitivamente se irá a implementar, por lo que al fin y al cabo, le es imposible evadir esta tarea. Sin embargo, con el proceso propuesto parte de esta tarea queda superada, ya que se ha adquirido aprendizaje previo y un conocimiento más completo de las limitaciones del problema, por tanto será mucho más sencillo para el decisor el análisis multiatributo y más aún cuando a esto se le suma que se cuenta con un número de alternativas mucho menor.

6.6 CONCLUSIONES

El algoritmo genético multiobjetivo propuesto (AGEM) se acerca mucho más a la frontera real para el problema de la mochila con 2 y 3 objetivos, en comparación a las metodologías NSGAI y SPEA2. Sin embargo, la distribución de sus individuos no es tan uniforme como la obtenida por SPEA2, pero si mucho más que la alcanzada por NSGAI aún cuando AGEM utiliza el mismo procedimiento de NSGAI para distribuir sus individuos.

Resultados de la simulación

Con el fin de mejorar esta distribución, se podría usar la metodología propuesta por SPEA2 y analizar los resultados.

Al estudiar el comportamiento durante varias corridas de NSGAI, SPEA2 y AGEM en el problema de la mochila, se observó que estos algoritmos genéticos multiobjetivo tienden a focalizar su búsqueda en la zona central de la frontera de Pareto, por tanto no se le podría ofrecer al decisor alternativas de solución ubicadas a los extremos de la frontera, las cuales podrían ser de interés para él. Orientando la búsqueda de forma interactiva se pudo solucionar este inconveniente, pues se obtuvo resultados que no eran posibles obtenerlos sin intervenir en el proceso.

Otra de las ventajas de AGEM-P, es que limitando la búsqueda en zonas de interés para el decisor, se pudo usar el mismo costo computacional en regiones más pequeñas logrando alcanzar niveles más cercanos a la frontera real, por lo cual se puede concluir que es conveniente involucrar las preferencias del decisor durante la búsqueda y de la manera en que lo hace AGEM-P.

7 CONCLUSIONES Y TRABAJOS FUTUROS

Después del desarrollo de esta investigación se puede concluir lo siguiente:

- La comunidad científica de los algoritmos genéticos multiobjetivo ha centrado su mayor atención en los algoritmos que hallan toda la frontera de Pareto y mejor distribuida, sin solucionar completamente el problema de decisión, puesto que con la solución encontrada con estos algoritmos se continuaría con un problema de decisión discreta.
- La metodología propuesta en esta investigación (AGEM) alcanza mejores resultados que los obtenidos por NSGAI y SPEA2 cuando está resolviendo el problema de la mochila para 2 objetivos. En el caso de tres objetivos no se alcanza a distinguir cual de las tres metodologías logra un mayor acercamiento a la frontera real, pero AGEM está al alcance de las otras dos metodologías que son reconocidas dentro del estado del arte.
- Con respecto a la distribución, AGEM cuando está resolviendo el problema de la mochila con 2 objetivos, obtiene la mayoría de las veces, individuos con una distribución más uniforme que los obtenidos con las otras dos metodologías. No ocurriendo lo mismo para el caso de tres objetivos, ya que SPEA2 tiene mejor distribución, sin embargo, si es mucho más uniforme que la alcanzada por NSGAI aún cuando AGEM utiliza el mismo procedimiento para distribuir sus individuos.
- Aún los algoritmos denominados referencia en el estado del arte (NSGA-II, SPEA2), aplicados al problema de la mochila, pueden arrojar soluciones que no comprendan las de interés para el decisor, ya que se encontró que estos algoritmos tienden a hallar todos sus individuos muy cercanos entre sí concentrando su búsqueda en una zona de la frontera de Pareto, que en este caso fue la central, olvidando las posibles soluciones ubicadas a los extremos y que podían ser de interés para el decisor.
- El uso de las preferencias en estas metodologías para el problema de la mochila, permite que el decisor pueda contemplar algunos de estos puntos como posible solución, ya que puede dirigir la búsqueda hacia esa zona.

Conclusiones y Trabajos futuros

- Las preferencias del decisor son cambiantes en el tiempo dependiendo el escenario en el que se encuentre, así como sus aspiraciones y deseos de logro varían en la medida que se aprende del problema, es por esto que es riesgoso considerar que las preferencias del decisor son absolutas y que una vez definidas él no estaría interesado en volverlas a cambiar. Por tanto, es necesario darle la opción al decisor modificar sus preferencias en la medida que va aprendiendo del problema y en el momento que el desee, condiciones que cumple AGEM-P.
- AGEM-P con su manera de incluir las preferencias, para el problema de la mochila, logra que el decisor pueda obtener alternativas que no son posibles con una metodología sin preferencias, permitiéndole realmente obtener soluciones conforme a sus intereses.
- Limitando la zona de búsqueda con las preferencias del decisor, permite que no se desperdicie costo computacional buscando al interior de una amplia zona cuando quizás las soluciones encontradas en ella serán descartadas por no ser del interés del decisor. Por tanto ese costo computacional AGEM-P lo utiliza para buscar soluciones más eficientes en la porción deseada de la frontera.
- En la medida que el algoritmo con preferencias avanza y encuentra soluciones cercanas a la frontera de Pareto, el espacio de búsqueda será cada vez más restringido, lo que lleva a que el decisor necesite modificar sus parámetros de exploración y explotación para que su búsqueda sea cada vez más eficiente, puesto que si continua con los establecidos inicialmente, el algoritmo perdería tiempo en buscar soluciones por fuera de la zona limitada por las restricciones.

TRABAJOS FUTUROS

Algunos de los trabajos que se podrían desarrollar después de esta investigación son:

- Proponer un nuevo concepto de distribución de los individuos para los algoritmos genéticos multiobjetivo que sea efectivo desde el inicio de la búsqueda, logrando que la población se disperse a lo largo de la zona limitada por las restricciones.
- Proponer tasas de cruzamiento y mutación autoajustables dependiendo de qué tan dispersa esté la población en determinada iteración.
- Construir nuevas métricas que determinen que tanto un algoritmo se acerca a la frontera real de Pareto y que se distribuya a lo largo de ella.

- Plantear una metodología donde el decisor pueda definir los sacrificios entre los objetivos mediante variables lingüísticas y que el concepto de dominancia sea considerado de una forma difusa.
- Hacer uso de técnicas de inteligencia artificial como redes neuronales o sistemas de inferencia difusos para entrenar un agente que tome las decisiones acerca de las preferencias durante el proceso de búsqueda del algoritmo genético multiobjetivo.
- Construir metodologías que tengan en cuenta múltiples decisores en el momento de definir las preferencias.
- Aplicar AGEM y AGEM-P a problemas reales, evaluando la satisfacción del decisor con respecto a las soluciones encontradas.
- Mejorar el aplicativo, de tal forma que se le pueda brindar al decisor la información cuando se esté resolviendo un problema con más de dos objetivos.

8 BIBLIOGRAFÍA

- [1] Abbass, H. A., Sarker, R., y Newton, C., "PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems," presented at Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, Seoul, South Korea, 2001.
- [2] Benayoun, R., Montgolfier, J., Tergny, J., y Laritchev, O., "Linear Programming with Multiple Objective Functions: Step Method (STEM)," *Mathematical Programming*, vol. 1, pp. 366-375, 1971.
- [3] Caballero, R., Luque, M., Molina, J., y Ruiz, F., "Programación Multiobjetivo Interactiva," vol. Monografía, No. 1, pp. 123 –148, 2002.
- [4] Charnes, A., Cooper, W., y Ferguson, R., "Optimal estimation of executive compensation by linear programming," *Management Science*, vol. 1, pp. 138-151, 1955.
- [5] Coelho, R. F., Bersini, H., y Bouillard, P., "Parametrical mechanical design with constraints and preferences: application to a purge valve," *Computer methods in applied mechanics and engineering*, vol. 192, pp. 4355–4378, 2003.
- [6] Coello, C. C., "An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends," *IEEE*, pp. 13, 1999.
- [7] Coello, C. C., "Handling Preferences in Evolutionary Multiobjective Optimization: A Survey," presented at Congress on Evolutionary Computation, 2000.
- [8] Coello, C. C., "Evolutionary Multi-Objective Optimization: A Critical Review," *Evolutionary Optimization*, 2002.
- [9] Coello, C. C., "Twenty Years of Evolutionary Multi-Objective Optimization: A Historical View of the Field," pp. 20 p., 2005.
- [10] Coello, C. C., "Optimización Evolutiva Multi-Objetivo: Pasado, Presente y Futuro," presented at Conferencia en la Universidad Nacional de Colombia, Medellín, 2007.
- [11] Coello, C. C., Veldhuizen, D. A. V., y Lamont, G. B., *Evolutionary Algorithms for Solving Multi-objective Problems*. New York: Plenum Publishers, 2002.
- [12] Coello, C. C., "Introducción a la Computación Evolutiva: Notas de Curso." México: Departamento de Ingeniería Eléctrica, Sección de Computación. Av. Instituto Politécnico Nacional, 2004, pp. 310.
- [13] Cohon, J. L. y Marks, D. H., "A Review and Evaluation of Multiobjective Programming Techniques," *Water Resources Research*, vol. 11(2), pp. 208–220, 1975.

Bibliografía

- [14] Cvetković, D. y Coello, C. A. C., "Human Preferences and their Applications in Evolutionary Multi-Objective Optimisation," in *RCS version 1.21*. Canadá. 2004, pp. 24.
- [15] Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*, Second edition ed. Chichester, UK: John Wiley & Sons, Inc., 2001.
- [16] Deb, K. y Goel, T., "Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence," in *Evolutionary Multi-Criterion Optimization*, vol. 1993/2001, Heidelberg, S. B., Ed., 2001, pp. 67-81.
- [17] Deb, K. y Chaudhuri, S., "I-EMO: An Interactive Evolutionary Multi-Objective Optimization Tool," in *Pattern Recognition and Machine Intelligence*, Heidelberg, S. B., Ed. Kanpur, India.: Kanpur Genetic Algorithms Laboratory (KanGAL), 2005, pp. 690-695.
- [18] Deb, K., Pratap, A., Agarwal, S., y Meyarivan, T., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182-197, 2002.
- [19] DeJong, K. A., "Analysis of the Behavior of a Class of Genetic Adaptive Systems," in *Computer and Communication Sciences Department*. Michigan: University of Michigan, 1975, pp. 256.
- [20] Fogel, L. J., Owens, A. J., y Walsh, M. J., "Artificial Intelligence through Simulated Evolution," John Wiley & Sons, I., Ed. New York, NY, USA, 1966.
- [21] Fonseca, C. M. y Fleming, P. J., "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Genetic Algorithms: Proceedings of the Fifth International Conference*, S. Forrest, e., Ed. San Mateo: Morgan Kaufmann, 1993, pp. 416-423.
- [22] Hillier, F. y Lieberman, G. J., *Investigación de Operaciones*, Séptima ed: McGrawHill, 2002.
- [23] Hillier, M. S. y Hillier, F. S., "Conventional Optimization Techniques," in *Evolutionary Optimization*, Sarker, R., Mohammadian, M., y Yao, X., Eds. California: Kluwer Academic Publishers, 2002, pp. 418.
- [24] Holland, J. H., "Concerning efficient adaptive systems," in *Self-Organizing Systems - 1962*, M. C. Yovits, Jacobi, G. T., y Goldstein., G. D., Eds. Washington D.C.: Spartan Books, 1962, pp. 215 - 230.
- [25] Holland, J. H., "Adaptation in Natural and Artificial Systems," University of Michigan Press, Michigan 1975.
- [26] Horn, J., Nafpliotis, N., y Goldberg, D. E., "A niched Pareto genetic algorithm for multiobjective optimization," *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on.*, vol. 1, pp. 82 - 87, 1994.
- [27] Ishibuchi, H., Nojima, Y., Narukawa, K., y Doi, T., "Incorporation of decision maker's preference into evolutionary multiobjective optimization algorithms.," presented at Genetic and Evolutionary Computation Conference (GECCO'2006), Washington, USA, 2006.

- [28] Jaramillo, G. P., "Desarrollo de un sistema soporte a la decisión para la asignación de recursos naturales con satisfacción de múltiples objetivos y múltiples decisores." España: Universidad Politécnica de Valencia, 1999.
- [29] Jin, Y. y Sendhoff, B., "Incorporation of Fuzzy Preferences into Evolutionary Multiobjective Optimization.," presented at Proceedings of the 4th Asia Pacific Conference on Simulated Evolution and Learning, Singapore, 2002.
- [30] Korhonen, P. y Laakso, J., "A Visual Interactive Method for Solving the Multiple Criteria Problem (VIA)," *European Journal of Operational Research*, vol. 24, pp. 277-287, 1986.
- [31] Marglin, J. A., "Public investment criteria," in *MIT Press*. Cambridge, Massachusetts, EE.UU., 1967.
- [32] Rachmawati, L. y Srinivasan, D., "Preference Incorporation in Multi-objective Evolutionary Algorithms: A Survey," presented at IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, 2006.
- [33] Rechenberg, I., "Cybernetic solution path of an experimental problem," *Royal Aircraft Establishment Library Translation no. I 122*, 1965.
- [34] Santana, L. V. y Coello, C. A. C., "Una introducción a la computación evolutiva y algunas de sus aplicaciones en Economía y Finanzas," *Revista de Métodos Cuantitativos para la economía y la empresa.*, vol. 2, pp. 3-26, 2006.
- [35] Schaffer, J. D., "Multiple objective optimization with vector-evaluated genetic algorithms," presented at 1st International Conference on Genetic Algorithms and their Applications, Carnegie-Mellon Univ., Pittsburgh, 1985.
- [36] Smith, R., Mesa, O., Dwyer, I., Jaramillo, P., Poveda, G., y Valencia, D., *Decisiones con Múltiples Objetivos e Incertidumbre*, Segunda Edición Ampliada ed. Medellín: Universidad Nacional de Colombia, 2000.
- [37] Srinivas, N. y Deb, K., "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," *Journal of Evolutionary Computation*, vol. 2. No. 3, pp. 221-248, 1994.
- [38] Veldhuizen, D. A. V. y Lamont, G. B., "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art," *Evolutionary Computation*, vol. 8, pp. 125-147, 2000.
- [39] Zadeh, L., "Optimality and non-scalar-valued performance criteria," *Automatic Control, IEEE Transactions*, vol. 8, pp. 59-60, 1963.
- [40] Zeleny, M., "Compromise Programming. en Multiple Criteria Decision Making," *University of South Carolina Press, Columbia, USA*, pp. 262-301, 1973.
- [41] Zitzler, E. y Thiele, L., "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions On Evolutionary Computation*, vol. 3. No. 4, pp. 257-271, 1999.
- [42] Zitzler, E., Laumanns, M., y Thiele, L., "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," Swiss Federal Institute of Technology (ETH), Zurich, Switzerland TIK-Report 103, May 2001.

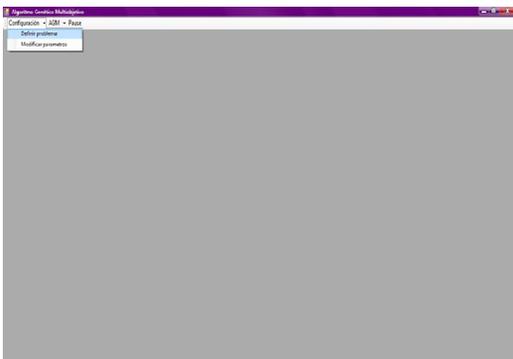
ANEXOS

ANEXO A. Manual de Usuario: Herramienta basada en AGEM-P para resolver el problema de la Mochila

Esta herramienta está diseñada para resolver el problema de la Mochila con 2 objetivos o más, con la metodología AGEM-P, donde el decisor, después de la decima parte de todas las iteraciones, puede intervenir en el proceso de búsqueda orientándola a encontrar individuos conforme a sus preferencias.

Además de esto, esta herramienta le permite visualizar el proceso cuando se está comparando las metodologías AGEM, NSGA-II y SPEA2.

Antes de Iniciar el proceso, es necesario definir el problema, para ello debe seleccionar “Configuración” y luego “Definir problema”, así:



En la siguiente pantalla, usted deberá ingresar los datos correspondiente al problema en estudio, tenga en cuenta que esos deben coincidir con los valores que están en el archivo que importará.

A screenshot of the 'Definición del problema' dialog box. It contains the following fields and controls:

- 'Número de variables:' and 'Número de objetivos:': Each with a text input field.
- 'Restricciones' section:
 - 'Nombre:': Text input field.
 - 'Tipo:': A dropdown menu with '≤' selected.
 - 'Capacidad:': Text input field.
 - 'Agregar' and 'Eliminar' buttons.
- A table with columns: 'Número', 'Nombre', 'Tipo', 'Capacit...'. The table is currently empty.
- 'Importar datos:': Text input field with an 'Examinar' button.
- 'Exportar datos a:': Text input field with an 'Examinar' button.
- 'Guardar' and 'Cancelar' buttons at the bottom.

En esta ventana usted deberá colocar el número de variables, el número de objetivos, el nombre de las restricciones y sus capacidades, al igual que deberá importar un archivo .txt donde estén los datos del problema, y seleccionar la ruta donde se exportarán los resultados. El archivo que contienen los datos, debe guardar un orden, tal como sigue:

Variable	Objetivo 1	Objetivo 2	Restricción 1	Restricción 2
1	148	113	106	100
2	64	138	17	68
3	101	87	18	34
4	127	18	54	103
5	141	91	17	93
6	120	111	52	49
7	29	39	56	18
8	97	73	87	125
9	144	89	71	121
10	33	124	148	131
11	70	73	129	126
12	21	50	69	86
13	48	32	76	66
14	88	32	67	125
15	88	100	118	70
16	83	58	115	146
17	44	15	108	55
18	127	149	30	142
19	109	27	145	77
20	75	17	150	26
21	112	67	109	51
22	90	132	144	76
23	61	68	23	121
24	134	55	135	132
25	76	144	35	27
26	146	46	95	12
27	38	106	131	14
28	66	143	118	127
29	57	60	12	21
30	84	106	21	96
31	49	13	136	18
32	124	104	76	36
33	46	150	110	119

Primera columna: número de las variables de decisión (x_i).

Segunda columna: coeficientes para el primer objetivo (c_{i1}).

Tercera columna: coeficientes para el segundo objetivo (c_{i2}).

Cuarta columna: coeficientes para la primera restricción (a_{i1}).

Quinta columna: coeficientes para la segunda restricción (a_{i2}).

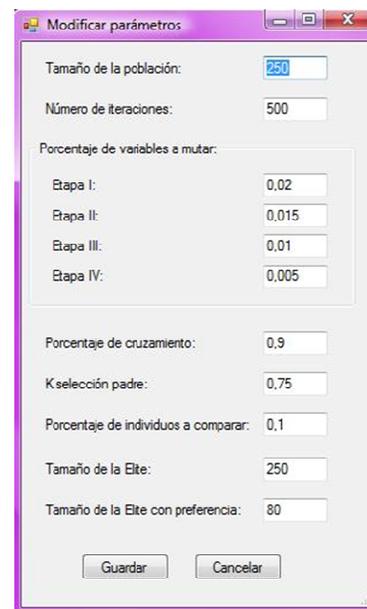
En el caso de resolver un problema con más de dos objetivos, debe colocar, luego del número de las variables de decisión, en las primeras columnas los coeficientes para las funciones objetivos y luego los coeficientes de las restricciones.

Recuerde que el número de variables de decisión, número de objetivos y número de restricciones que tenga dentro del archivo debe coincidir con los datos que usted ingresó en la ventana de "Definición del problema".

La herramienta tiene varios archivos como ejemplo que pueden ser importados. El nombre de cada uno de ellos especifica el número de variables y

objetivos que considera. Por ejemplo, datos KP4002 son los datos correspondientes al problema de la mochila para 400 variables de decisión y 2 objetivos.

Si usted desea, antes de correr AGEM-P, puede modificar los parámetros del algoritmo en Configuración → Modificar parámetros.



En esta ventana usted podrá modificar el tamaño de la población con que trabaja el algoritmo, el número de iteraciones, los porcentajes de variables a mutar, el porcentaje de cruzamiento, el parámetro K de la selección por torneo para escoger los padres, el porcentaje de individuos a comparar cuando se hace la selección de los padres, el tamaño de la elite con que trabajará AGEM y el tamaño de la elite con que trabajará AGEM-P.

En el caso de las variables a mutar, el porcentaje varía por etapa cuando corre el método AGEM, es decir, en la comparación de metodologías, donde una etapa corresponde a la quita parte del total de iteraciones y las dos últimas quinta parte corresponden a la etapa IV y tienen el mismo valor de variables a mutar. Recuerde que cuando hace uso de las preferencias, AGEM-P usa sólo el porcentaje de variables a mutar de la primera etapa, ya que después el decisor podrá modificarlo durante las corridas sin necesidad de volver a esta ventana.

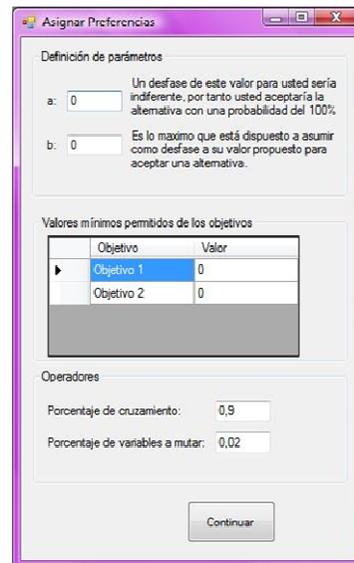
Luego de definir el problema y cambiar parámetros si lo desea, usted podrá decidir si quiere resolver el problema con AGEM-P o si desea comparar la metodología propuesta (AGEM) con NSGAI y SPEA2. Esto lo define en AGM → Usar Preferencias con AGEM-P ó Comparar metodologías.

Mientras corre el algoritmo, con ayuda del mouse, usted podrá aumentar o disminuir la escala de las gráficas haciendo un clic sostenido en ella y rodeando la zona que desea aumentar, lo mismo que puede ubicar las gráficas en la pantalla como mejor le parezca con el fin de visualizar toda la información, para

esto debe hacer clic en la parte superior de la ventana y arrastrarla al lugar que desea.

Si está haciendo uso de AGEM-P usted podrá parar el algoritmo cuando desee, dando clic en Pause, ubicado en la parte de arriba de la ventana principal.

Cuando esto sucede, le aparecerá una ventana donde usted debe definir sus parámetros de preferencia y los valores mínimos permitidos para usted en cada objetivo. Además de ello, podrá modificar los parámetros de cruzamiento y mutación dependiendo si desea tener una mayor exploración y/o explotación.



ANEXO B. Herramienta Interactiva para resolver el problema de la mochila con el método AGEM – P

La herramienta interactiva se encuentra en el CD anexo a este documento.

ANEXO C. Datos para el problema de la Mochila

Los datos utilizados para resolver el problema de la mochila se encuentran anexos a este documento en el CD.