

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN



**ADAPTACIÓN DE UN ALGORITMO DE AGRUPACIÓN  
MONO-OBJETIVO A MULTI-OBJETIVO USANDO  
CARACTERIZACIÓN DEL DESEMPEÑO**

**TESIS**

**QUE PARA OBTENER EL GRADO DE:  
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN**

**PRESENTA**

**I. S. C. ENITH MARTÍNEZ CRUZ**

**G09071230**

**DIRECTOR DE TESIS:**

**DRA. LAURA CRUZ REYES**



"Año del Centenario de la Promulgación de la Constitución Política de los Estados Unidos Mexicanos"

Cd. Madero, Tamps; a 22 de Mayo de 2017

OFICIO No.: US.092/17  
AREA: DIVISIÓN DE ESTUDI  
DE POSGRADO E INVESTIGACIÓN  
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS

**ING. ENITH MARTÍNEZ CRUZ**  
**NO. DE CONTROL G09071230**  
**PRESENTE**

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su examen de grado de Maestría en Ciencias de la Computación, el cual está integrado por los siguientes catedráticos:

PRESIDENTE :	DRA. MARÍA LUCILA MORALES RODRÍGUEZ
SECRETARIO :	DRA. CLAUDIA GUADALUPE GÓMEZ SANTILLÁN
VOCAL :	DRA. LAURA CRUZ REYES
SUPLENTE	DR. NELSON RANGEL VALDEZ
DIRECTOR DE TESIS:	DRA. LAURA CRUZ REYES

Se acordó autorizar la impresión de su tesis titulada:

**"ADAPTACIÓN DE UN ALGORITMO DE AGRUPACIÓN MONO-OBJETIVO A MULTIOBJETIVO USANDO CARACTERIZACIÓN DEL DESEMPEÑO"**

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con Usted el logro de esta meta.

Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

**ATENTAMENTE**  
"POR MI PATRIA Y POR MI BIEN"®

**DRA. ADRIANA ISABEL REYES DE LA TORRE**  
**JEFA DE LA DIVISIÓN**



c.c.p.- Archivo  
Minuta

AIRTELICO jar  
M



# Declaración de originalidad

Yo, Enith Martínez Cruz, en mi calidad de autor manifiesto que este documento de tesis es producto de mi trabajo original y que no infringe los derechos de terceros, tales como derechos de publicación, derechos de autor, patentes y similares.

Además, declaro que en las citas textuales que he incluido (las cuales aparecen entre comillas) y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y publicaciones de referencia.

En caso de presentarse cualquier reclamación o acción por parte de un tercero en cuanto a los derechos de autor sobre la obra en cuestión, asumiré toda la responsabilidad y relevo de ésta a mi director y codirectores de tesis, así como al Instituto Tecnológico de Ciudad Madero y sus autoridades.

Cd. Madero, Tamaulipas, junio de 2017.

# Resumen

En la actualidad en diferentes áreas es muy fácil encontrarse con problemas de agrupación, diferentes investigadores han abordado este tipo de problemas, y en su mayoría se han obtenido resultados satisfactorios, pero la complejidad de resolver este tipo de problemas aumenta cuando se trata de dar solución al problema tomando en cuenta más de un objetivo, lo que conocemos en computación como optimización multi-objetivo, ya que al tratar con múltiples objetivos debe asegurarse que la solución seleccionada no descarte o menosprecie a un objetivo. Además, es importante mencionar que al momento de resolver un problema multi-objetivo no se genera una solución sino un conjunto de soluciones por lo que es necesario explorar cada una de esas soluciones para encontrar aquella solución óptima o una solución que se acerque al óptimo, esta no es una tarea fácil, por eso a la complejidad de este tipo de problemas de optimización se cataloga como NP-Duro.

Particularmente, para tratar con la complejidad de problemas de optimización difíciles NP-Duros, se han desarrollado infinidad de algoritmos heurísticos, sin embargo, pocos de estos trabajos describen características de los algoritmos, de manera tal que permitan explicar su comportamiento ante ciertas situaciones. Estos algoritmos están enfocados a resolver cierto tipo de instancias y cuando se desea experimentar con otro conjunto de instancias el desempeño del algoritmo puede no ser el deseado. Entre los algoritmos que se encuentran en la literatura para resolver un problema de agrupación mono-objetivo está el desarrollado por Quiroz (Quiroz, 2014) el cual está enfocado a resolver el problema de empaclado en contenedores de una dimensión haciendo uso de métricas de caracterización para evaluar el desempeño de dicho algoritmo. El problema que resuelve es un problema mono-objetivo por lo que en esta tesis se realizó la adaptación de dicho algoritmo para que resuelva un problema multi-objetivo de agrupación, los resultados de la experimentación revelan que la nueva versión denominada M-GGA-CGT supera el rendimiento del algoritmo del estado del arte. NSGA2

# CONTENIDO

---

<b>I. INTRODUCCIÓN.....</b>	<b>6</b>
1.1 Antecedentes del proyecto.....	7
1.2 Descripción del problema de investigación.....	9
1.3 Justificación.....	9
1.4 Objetivos del proyecto.....	10
1.4.1 Objetivo general.....	10
1.4.2 Objetivos específicos.....	11
1.5 Alcances y limitaciones.....	11
1.6 Contenido de la tesis.....	11
<b>II. MARCO TEÓRICO.....</b>	<b>13</b>
2.1 Problema multi-objetivo Path Dissimilarity.....	13
2.2 Caracterización.....	15
2.2.1 Análisis de datos exploratorio.....	15
2.2.2 Análisis de la estructura del espacio de búsqueda del problema.....	16
2.2.3 Estudio de parámetros que ocasionan transiciones de fases en la dificultad del problema16	
2.2.4 Análisis de la estructura de las soluciones óptimas.....	17
2.3 Optimización multi-objetivo.....	17
2.3.1 Variables de decisión.....	18
2.3.2 Restricciones.....	18
2.3.3 Modelo de un problema de optimización multi-objetivo.....	19
2.3.4 Frente de Pareto.....	19
2.4 Métricas de desempeño.....	20
2.4.1 Distancia generacional.....	21
2.4.2 Distancia Generacional Invertida (IGD).....	21
2.4.3 Spread.....	22
2.4.4 Tasa de errores.....	22
2.4.5 Hipervolumen (HV).....	23
2.4.6 Indicador de Dominancia de Pareto (NR).....	24
2.4.7 Cobertura de conjuntos.....	24
2.4.8 Distribución Uniforme (UD).....	25

2.4.9	Espaciado (S) .....	25
2.4.10	Máxima amplitud (MS).....	25
2.4.11	Análisis de métricas de desempeño.....	26
<b>III.</b>	<b>ESTADO DEL ARTE .....</b>	<b>28</b>
3.1	Obtención de explicaciones del desempeño algorítmico .....	28
3.2	Obtención de explicaciones del desempeño algorítmico .....	29
3.2.1	Medidas de caracterización .....	29
3.2.2	Revisión de trabajos de la literatura .....	32
3.2.3	Análisis comparativo de trabajos relacionados.....	33
<b>IV.</b>	<b>ADAPTACIÓN DEL ALGORITMO GGA-CGT .....</b>	<b>36</b>
4.1	Estructura original del algoritmo .....	36
4.2	Rediseño de la estructura del algoritmo .....	37
4.2.1	Componentes del algoritmo M-GGA-GCT .....	38
4.3	Configuración de parámetros del algoritmo M-GGA-GCT .....	43
4.3.1	Metodología propuesta para la configuración de parámetros .....	44
<b>V.</b>	<b>EXPERIMENTACIÓN Y RESULTADOS.....</b>	<b>47</b>
5.1	Evaluación inicial de M-GGA-CGT .....	47
5.2	Caracterización del desempeño .....	49
5.2.1	Caracterización del Problema .....	49
5.2.2	Caracterización de la salida .....	52
<b>VI.</b>	<b>CONCLUSIONES .....</b>	<b>57</b>
6.1	CONCLUSIONES .....	57
6.2	TRABAJO FUTURO .....	58
<b>VII.</b>	<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>59</b>

# Capítulo 1

## I. INTRODUCCIÓN

---

En la actualidad existen infinidad de problemas que tienen como respuesta un número exponencial de posibles soluciones por lo que es necesario explorar cada una de ellas para encontrar la solución óptima o una que se acerque al óptimo. Es posible seleccionar cualquiera de esas soluciones factibles, pero en la vida real se trata de elegir la mejor solución, según sea el caso: la que minimice los costos, la que maximice ganancias, o alguna otra acción de mejora. Puede decirse que este es un proceso en donde se selecciona, de lo posible, lo mejor. Al proceso de tratar de encontrar la mejor solución posible para un determinado problema se le denomina optimización aproximada ya que no garantizan encontrar el óptimo.

El proceso para seleccionar una solución se vuelve casi imposible cuando se tienen múltiples objetivos a alcanzar. Es decir, cuando se quiere lograr una meta, pero tomando en cuenta varios objetivos al mismo tiempo. Ahí lo que se desea es encontrar aquella solución que establezca un balance entre los criterios, a esto se le denomina Optimización Multi-objetivo.

Particularmente, para tratar con la complejidad de problemas de optimización difíciles NP-Duro, se han desarrollado infinidad de algoritmos aproximados, a los que se denomina también algoritmos heurísticos, unos más complicados de abordar que otros. Sin embargo, pocos de estos trabajos describen características de los algoritmos, de manera tal que permitan saber en qué momento es conveniente usar cada uno de ellos. Entre los algoritmos que se encuentran en la literatura para resolver este tipo de problemas está el desarrollado por (Quiroz, 2014) el cual está enfocado a resolver el problema de agrupación mono-objetivo, en donde haciendo uso de métricas de caracterización evalúa el desempeño algorítmico.

En este trabajo se aplica la metodología propuesta por (Quiroz, 2014) mediante una adaptación algorítmica en donde se resuelve un problema de agrupación multi-objetivo.

## **1.1 ANTECEDENTES DEL PROYECTO**

Este trabajo se plantea en el contexto de un proyecto de investigación sobre algoritmia experimental llevado a cabo en la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Cd. Madero. En particular se tiene interés en estudiar el comportamiento de un algoritmo que dé solución a un problema de agrupación multi-objetivo.

En el trabajo de Cruz (Cruz, 2004) se realiza la caracterización a un conjunto de algoritmos con la finalidad de crear modelos de predicción, pudiendo determinar qué algoritmo ofrece un mejor desempeño al resolver una instancia en particular.

Santiago aplica la metodología de selección al problema DFAR (Santiago, 2004). Landero (Landero, 2004) forma grupos de dominación mediante la exploración de varias técnicas de clasificación y agrupación; lo anterior permitió asociar un grupo de instancias con el algoritmo que mejor las resuelve.

García (García, 2004) incorpora retroalimentación a la arquitectura de selección y caracteriza el comportamiento algorítmico utilizando índices de superficie de aptitud tomados de la literatura.

Álvarez (Álvarez, 2006) incorpora nuevos índices de caracterización para el problema de empaclado en contenedores obtenidos mediante estadística descriptiva. Luego, Ferat (Ferat, 2007) desarrolla un sistema de clasificación basado en redes neuronales para la selección dinámica y autoadaptiva de algoritmos heurísticos.

Más adelante, Pérez (Pérez, 2007) presenta una metodología para el análisis experimental de algoritmos heurísticos donde incorpora algunas técnicas de análisis multivariadas y modelado causal.

Landero (Landero, 2008) propone una metodología que incorpora un conjunto mayor de métricas de caracterización y muestra la aplicabilidad del modelado causal en la selección de algoritmos.

García (García, 2010) muestra la aplicabilidad de la caracterización del comportamiento algorítmico al desarrollar dos índices de la intensidad del uso de heurísticas dentro de un algoritmo hiper-heurístico.

Quiroz (Quiroz, 2014) caracterizó un algoritmo genético para dar solución al problema de empaqueo de objetos en contenedores (BPP, por su sigla en inglés). Identificó factores que influyen en el desempeño y se cuantifican mediante índices. Además, identificó propiedades que definen la estructura de una instancia de BPP y explicó el desempeño de los algoritmos. Generó nuevos conjuntos de instancias difíciles y diseñó nuevas estrategias inteligentes que fueron integradas en un algoritmo denominado GGA-CGT.

El análisis de los trabajos de investigación descritos en los párrafos anteriores, se observa que la calidad del conocimiento obtenido depende principalmente del poder de caracterización y de la validez de las métricas de caracterización propuestos. En el trabajo aquí propuesto, se aplican un enfoque de análisis experimental para obtener mayor conocimiento sobre el comportamiento del algoritmo GGA-CGT sobre un problema de optimización con dos objetivos. Se seleccionó este algoritmo porque hasta nuestro conocimiento es el mejor algoritmo reportado en la literatura para problemas de agrupación con un solo objetivo.

## 1.2 DESCRIPCIÓN DEL PROBLEMA DE INVESTIGACIÓN

El problema de explicar formalmente porqué un algoritmo de agrupación multi-objetivo sigue cierto comportamiento al resolver un conjunto de instancias de un problema multi-objetivo se puede describir de la siguiente manera. Esta descripción se adaptó de las formulaciones presentadas en [(Cruz, 2004), (Quiroz, 2014)].

Dados:

- Un algoritmo de agrupación multi-objetivo  $A$  para resolver un problema multi-objetivo  $P$
- Un conjunto de instancias  $I = \{i_1, i_2, i_3, \dots, i_n\}$  del problema  $P$ .

Se busca:

- Un conjunto de indicadores  $F = \{f_1, f_2, f_3, \dots, f_m\}$  que caractericen factores del problema y del algoritmo que definan el proceso de optimización algorítmico.
- Un conjunto de relaciones  $R = \{r_1, r_2, r_3, \dots, r_p\}$ , para  $1 \leq p \leq m$ , tal que cada  $r_i$  establece una relación causal entre los elementos de  $G_i$ , donde  $G_i \subseteq F$  que incluye factores que impactan el desempeño algorítmico.

Aplicar el conocimiento adquirido a través de relaciones causales  $R$ , al diseño de mejores estrategias de solución.

En otras palabras, se busca un conjunto de factores críticos que caractericen el proceso de optimización algorítmico y permitan explicar por qué el algoritmo objeto de estudio sigue cierto comportamiento al resolver un problema de optimización multi-objetivo y aplicar el conocimiento adquirido en:

1. Instancias representativas del problema.
2. El diseño de un algoritmo de alto desempeño.

## 1.3 JUSTIFICACIÓN

Los investigadores han desarrollado algoritmos cada vez más sofisticados para resolver problemas de optimización. Se han llevado a cabo estudios experimentales

para determinar qué algoritmos funcionan mejor, pero generalmente estos algoritmos están limitados a cierto conjunto de instancias así que al experimentar con otro conjunto el algoritmo no obtiene el mismo desempeño.

El descubrimiento de las condiciones en que un algoritmo de optimización tendrá éxito o no, es fundamental para la comprensión de las fortalezas y debilidades del algoritmo, y esto ayudará a la hora de seleccionar el algoritmo adecuado a la hora de resolver un problema.

Dentro de la literatura especializada se pueden encontrar gran cantidad de investigaciones en donde han trabajado con la caracterización de algoritmos mono-objetivo, pero existen pocos trabajos que aborden caracterización con algoritmos multi-objetivo. Por lo que en este trabajo se realizará la adaptación de un algoritmo que resuelve un problema mono-objetivo a multi-objetivo, con la finalidad de que esta adaptación permita al algoritmo alcanzar un alto desempeño en la solución de un problema de agrupación multi-objetivo.

Además se espera que este trabajo contribuya a promover la aplicación de técnicas empíricas de análisis dentro de las ciencias de la computación, facilitando a futuras investigaciones la obtención de teorías que fundamenten los resultados y comportamientos observados en algoritmos heurísticos que solucionan problemas de optimización multi-objetivo.

## **1.4 OBJETIVOS DEL PROYECTO**

### **1.4.1 Objetivo general**

Desarrollar un algoritmo heurístico competitivo que dé solución a un problema multi-objetivo de agrupación, cuyo alto desempeño sea posibilitado a través de la aplicación de técnicas de caracterización del proceso de optimización algorítmico.

### **1.4.2 Objetivos específicos**

1. Identificar métricas representativas del estado del arte que permitan caracterizar el proceso de optimización de un algoritmo heurístico que dé solución a un problema de agrupación multi-objetivo.
2. Identificar métodos de análisis experimental que permitan explicar el proceso de optimización de un algoritmo heurístico que dé solución a un problema de agrupación multi-objetivo.
3. Adaptar un algoritmo mono-objetivo de agrupación del estado del arte para la solución de un problema multi-objetivo de agrupación relacionado.

### **1.5 ALCANCES Y LIMITACIONES**

1. La caracterización del proceso de optimización se aplicará sólo a un problema de agrupación relacionado con el problema empacado de objetos de contenedores de una dimensión 1D-BPP y algoritmos de solución del estado del arte relevantes para este problema.
2. Las características identificadas pudieran ser un subconjunto de todas las que afectan el desempeño.

### **1.6 CONTENIDO DE LA TESIS**

Capítulo 2: Marco Teórico. En este capítulo se presenta las bases teóricas relacionadas con este trabajo de tesis, como lo es el problema multi-objetivo de los caminos diversos (Path Dissimilarity Problem, PDP), los conceptos principales de multi-objetivo y teoría sobre técnicas de caracterización.

Capítulo 3: Estado del Arte. Inicia describiendo trabajos sobresalientes que dan solución al problema multi-objetivo path dissimilarity. Posteriormente, se muestra una reseña de los trabajos relacionados con la caracterización del desempeño de algoritmos.

Capítulo 4: Adaptación del algoritmo genético del estado del arte GGA-CGT. Se detalla la estructura del algoritmo original, así como cada la adaptación y cada componente del algoritmo adaptado.

Capítulo 5: Experimentación y resultados. En este capítulo se presenta el desempeño obtenido del algoritmo M-GGA-CGT aplicado al problema multi-objetivo Path Dissimilarity comparando los resultados con el algoritmo del estado del arte NSGA-II.

Capítulo 6: Conclusiones. En este capítulo se presentan las conclusiones del trabajo de tesis, los productos de investigación, así como propuestas de trabajos futuros.

# Capítulo 2

## II. MARCO TEÓRICO

---

En este apartado se detallan los conceptos más importantes para abordar la caracterización de algoritmos heurísticos. Primero se tratan los enfoques que existen en el área de la caracterización y más adelante se analizan aquellos conceptos relacionados con los problemas multi-objetivo de agrupación.

### 2.1 PROBLEMA MULTI-OBJETIVO PATH DISSIMILARITY

El problema multi-objetivo Path Dissimilarity (PDP, por su siglas en inglés), es un problema de rutas, el cual consiste en generar  $P$  rutas desde un origen a un destino de manera tal que el promedio de la suma de las rutas sea la mínima y que la disimilitud entre las rutas sea la máxima (Martí, 2009).

El proceso de encontrar rutas o caminos diferentes en un grafo es un problema de optimización clásico y existen diversas opciones para generar las rutas. Nosotros consideramos que un elemento, en este caso un vértice, debe estar sólo en una ruta. Esta variante puede comprenderse en el siguiente escenario: se tiene material peligroso que debe ser transportado en camiones desde un punto de inicio a un punto final dentro de una ciudad.

Lo que se busca es que las rutas que tomen los diferentes elementos de transporte sean, en lo posible, más espaciadas dentro de la ciudad de tal manera que no existan tantos cruces entre las rutas, reduciendo así el riesgo del material peligroso, además, que la distancia de todos los elementos de transporte sea la mínima.

A continuación se describen los elementos del problema: dado un grafo no dirigido  $G(V, E)$  donde  $V$  es el conjunto de vértices y  $E$  es el conjunto de aristas con costo asociado  $c_{ij}$  en  $E$ , que generalmente es la distancia Euclidiana, y un par de vértices  $(o, d)$  que son origen y destino, respectivamente. Se define  $P(o, d)$  como el conjunto de todas las rutas en  $G$  desde  $o$  a  $d$ . Dado un número entero  $p > 1$ , una solución factible para el problema PDP es un conjunto  $S$  subconjunto de  $P(o, d)$  tal que  $|S| = p$ . Dada una solución  $S = \{P_1, P_2, \dots, P_p\}$ , se define su valor  $f_1(S)$  como el promedio de los costos de los caminos en  $S$  (Martí, 2009):

$$f_1(S) = \frac{\sum_{t=1}^p c(P_t)}{p} \quad \text{donde} \quad c(P_t) = \sum_{(i,j) \in P_t} c_{ij}$$

El valor de disimilitud  $f_2(S)$  es definido como el promedio de la disimilitud entre los distintos pares de caminos en  $S$ :

$$f_2(S) = \frac{\sum_{i=1}^{p-1} \sum_{j=i+1}^p \text{dis}(P_i, P_j)}{\binom{p}{2}}$$

Donde la disimilitud entre dos caminos  $P_i = \{v_1, v_2, v_3, \dots, v_n\}$  y  $P_j = \{u_1, u_2, u_3, \dots, u_m\}$  está dado por la promedio de las distancias entre cada vértice del camino  $P_i$  a  $P_j$  más el promedio de las distancias entre cada vértice en  $P_j$  a la ruta  $P_i$ . A partir de la distancia entre dos puntos, se define la distancia de un vértice  $v \in V$  a una ruta  $P_i = \{v_1, v_2, v_3, \dots, v_n\}$  de la siguiente manera:

$$\delta(v, P_i) = \min_{v_j \in P_i} \delta(v, v_j)$$

Quedando la medida de disimilitud como se muestra en la siguiente ecuación:

$$\text{dis}(P_i, P_j) = \frac{1}{2} \left( \frac{\sum_{v_i \in P_i} \delta(v_i, P_j)}{|P_i|} + \frac{\sum_{u_j \in P_j} \delta(u_j, P_i)}{|P_j|} \right)$$

El problema queda formulado de la siguiente manera:

$$\begin{aligned} (PDP) \quad & \min \quad f_1(S) \\ & \max \quad f_2(S) \\ & \text{sujeto a } S \subseteq P(o, d) \\ & |S| = p \end{aligned}$$

En este trabajo, el PDP es tratado como un problema de agrupación en donde los grupos son conformados por los vértices del grafo, de tal manera que una ruta es un grupo. A diferencia de (Martí, 2009), un vértice no puede estar en dos grupos a la vez, a excepción del vértice origen y el vértice destino. Para formar los grupos se considera que cada vértice tiene un peso, en este caso es la distancia entre vértices, y a cada grupo se le asigna un límite de capacidad (tamaño máximo de la ruta) acorde al tamaño de la instancia del problema.

## 2.2 CARACTERIZACIÓN

Las características son todos los aspectos que definen un objeto ya sea físicamente o lógicamente incluyendo su comportamiento en el medio en que se desenvuelve. Mientras más se conocen y comprenden las características de un objeto más fácil será trabajar con él. De ahí se desprende el concepto de caracterización que enfocado a la computación puede decirse que es el proceso en el cual se identifican todos los factores que intervienen en la solución de un problema, desde la estructura de las instancias, así como el comportamiento del algoritmo a usar y la estructura de las soluciones obtenidas (Quiroz, 2014).

### 2.2.1 Análisis de datos exploratorio

John W. Tukey dice (Tukey, 1977):

*“El análisis exploratorio de datos es, básicamente, el tratamiento estadístico al que se someten las muestras recogidas durante un proceso de investigación en cualquier campo científico”.*

Teniendo en cuenta la definición de John Tukey, en el campo computacional este tipo de análisis consiste en estudiar los datos de manera exhaustiva mediante diversos métodos estadísticos con la finalidad de comprender el comportamiento de la información obtenida en la experimentación de algoritmos desarrollados y al final tener mayor precisión al generar una hipótesis. Algunos de estos métodos usados en la caracterización de datos y algorítmico se reportan en (Bartz-Beislstein, 2014), (Veček, 2014), (Bielza, 2014), (Krause, 2011), (Camacho, 2015).

### **2.2.2 Análisis de la estructura del espacio de búsqueda del problema**

Un proceso de búsqueda en el área de la heurística permite conocer ampliamente los resultados obtenidos a partir de la función objetivo. El análisis del espacio de búsqueda, conocido en inglés como fitness Landscape y representado por estos resultados permite comprender el comportamiento del algoritmo y puede servir para aportar nuevas estrategias en el desarrollo del mismo. Los aspectos que se toman en cuenta en este tipo de estudio son (Duarte, 2007):

- Rugosidad: es la diferencia entre el valor de la función objetivo y dos soluciones cercanas.
- Picos: son los óptimos locales que arroja la función objetivo los cuales aparecen en la hiper-superficie.
- Distribución: analiza el cómo está esparcido cada óptimo local dentro del espacio de búsqueda.

### **2.2.3 Estudio de parámetros que ocasionan transiciones de fases en la dificultad del problema**

En muchos problemas complejos de optimización combinatoria, una transición de fase ocurre en términos de la facilidad/dificultad del problema. La dificultad para resolver un problema incrementa hasta un tamaño de  $n$  dado, y más allá de este valor es fácil de resolver. Por esta razón, los problemas más difíciles tienden a estar

en los límites de la transición de fase. El fenómeno de transición de fase sólo ha sido identificado en algunos problemas (Talbi, 2009).

Cuando se desarrolla un algoritmo se toman en cuenta todos los parámetros que definen a un problema con el fin de representarlo lo más realista posible. Por lo que una vez teniendo los resultados de la experimentación se puede explorar con los diferentes valores que se le asignan a cada parámetro, con la intención de conocer cómo se comporta el algoritmo en todos los casos, y qué valores permiten llegar al óptimo. Este tipo de estudio se enfoca en analizar los parámetros del problema para determinar ¿en qué momento? éstos provocan una *transición de fase*. Como ya se mencionó antes, en optimización podemos definir el cambio de fase como aquel cambio abrupto en la complejidad del problema (Quiroz, 2014)

#### **2.2.4 Análisis de la estructura de las soluciones óptimas**

Este estudio permite entender la relación que existe entre los valores de las soluciones y la dificultad de cada instancia (Quiroz, 2014).

Cuando se trata de heurísticas estas siempre generan  $n$  cantidad de soluciones, según le indique el desarrollador, cada solución se comporta de diferente manera. Unas se acercan más al óptimo otras en cambio se alejan, este comportamiento va construyendo lo que se conoce como Espacio de Búsqueda o superficie de soluciones. Muchos investigadores dedican su tiempo a estudiar el comportamiento que tienen las soluciones, estudian la estructura que éstas conforman pudiendo así interpretar en qué punto el algoritmo arroja mejores soluciones además de permitirle comprender aún más el algoritmo utilizado.

### **2.3 OPTIMIZACIÓN MULTI-OBJETIVO**

En la vida real se involucra la toma de decisiones donde es natural que siempre se desean tomar las mejores decisiones, en otras palabras, que estas decisiones sean

óptimas. Existen áreas en donde la decisión no se debe tomar al azar, sino que se deben emplear modelos matemáticos que apoyen la decisión tomada. Ya que para tomar una decisión se contempla no solo un objetivo o criterio, sino varios, por lo tanto, no se puede centrar la atención solo en uno de los objetivos dejando a un lado los otros. Esto es conocido como optimización multi-objetivo (Baesler, 2008).

La optimización multi-objetivo puede ser definida como un problema de decisión que presenta más de una función objetivo, este tipo de problemas no tiene una solución óptima única sino que genera un conjunto de soluciones óptimas, a este conjunto, expresado en el espacio de los valores de la función objetivo, se le denomina *Frente de Pareto* (Baesler, 2008).

### 2.3.1 Variables de decisión

Cuando se desea tratar un problema multi-objetivo es necesario seleccionar valores numéricos los cuales se denominan variables de decisión ya que sirven para dar solución al problema. De esta manera el vector, comúnmente nombrado vector solución, de  $n$  variables de decisión  $\vec{x}$  se representa de la siguiente manera (Coello, 2007):

$$\vec{x} = [x_1, x_2, \dots, x_n]^T$$

### 2.3.2 Restricciones

Para desarrollar un problema multi-objetivo es importante tener límites los cuales permiten identificar qué alcances puede tener el algoritmo, estos son conocidos como restricciones los cuales matemáticamente se representan de la siguiente manera (Coello, 2007):

$$g_i(\vec{x}) \leq 0; i = 1, 2, \dots, p$$

$$h_i(\vec{x}) = 0; i = 1, 2, \dots, q$$

### 2.3.3 Modelo de un problema de optimización multi-objetivo

Un problema de optimización multi-objetivo está definido de la siguiente manera (Coello, 2007):

$$\min/\max : f_m(\vec{x}), m = 1, 2, \dots, M$$

s. a.

$$g_j(\vec{x}) \geq 0, j = 1, 2, \dots, p$$

$$h_k(\vec{x}) = 0, k = 1, 2, \dots, q$$

$$x_i^l \leq \vec{x} \leq x_i^u, i = 1, 2, \dots, n$$

Donde  $\vec{x} \in \mathbb{R}^n$  se define como el vector de decisión,  $f_m(\vec{x}) \in \mathbb{R}^n$  como el vector de objetivos y  $g_j(\vec{x})$ ,  $h_k(\vec{x})$  como vectores de restricciones de desigualdad e igualdad, respectivamente. Finalmente,  $\vec{x}_i^l$  y  $\vec{x}_i^u$  representan las cotas inferiores y superiores del espacio de decisión.

### 2.3.4 Frente de Pareto

Para trabajar con un problema de optimización multi-objetivo se debe definir algún tipo de evaluación con el fin de determinar qué soluciones se considera que aportan mejor beneficio que otras, para lo que se introduce el concepto de dominancia. El cual consiste en clasificar las soluciones y encontrar buenas alternativas que contemplen todos los objetivos a optimizar.

Por lo tanto se dice que una solución  $\vec{x}^1$  domina a otra solución  $\vec{x}^2$  si se cumplen las siguientes condiciones (Coello, 2007):

1. La solución  $\vec{x}^1$  no es peor que la solución  $\vec{x}^2$  en todos los objetivos.
2. La solución  $\vec{x}^1$  es estrictamente mejor que la solución  $\vec{x}^2$  en por lo menos un objetivo.

Para un problema multi-objetivo dado, si  $P$  representa todo el espacio de soluciones posibles, entonces al conjunto todas las soluciones de  $P$  que no son dominados por otras soluciones de  $P$  se le conoce como conjunto óptimo de Pareto. A la región de puntos definida por el conjunto óptimo de Pareto en el espacio de valores de las funciones se conoce como frente de Pareto. De esta manera el frente de Pareto se estará conformado solo de soluciones no dominadas.

En la Figura 1 se muestran los conjuntos óptimos de Pareto para diferentes escenarios con dos objetivos y para el mismo espacio de soluciones. En la gráfica del inciso a se muestra el frente de Pareto cuando se desea minimizar los dos objetivos, en el inciso b se muestra el frente óptimo cuando se desea maximizar el primer objetivo y minimizar el segundo objetivo, en la gráfica del inciso c se muestra el óptimo de Pareto cuando el primer objetivo es minimizado y el segundo es maximizado, en la gráfica del inciso d se muestra cuando los dos objetivos quieren ser maximizados.

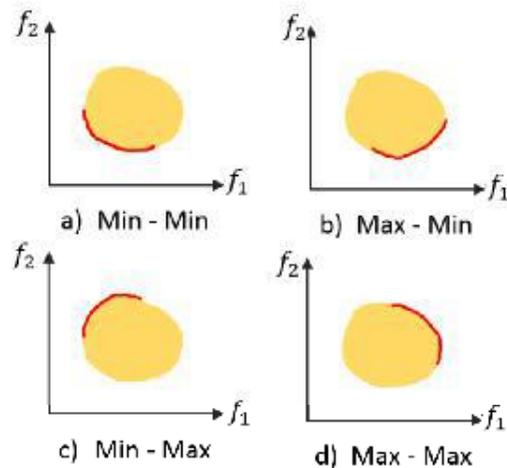


Figura 1. Frentes Óptimos de Pareto para el mismo espacio de soluciones.

## 2.4 MÉTRICAS DE DESEMPEÑO

Para evaluar el rendimiento de los algoritmos se debe tener en cuenta dos cuestiones diferentes: reducir al mínimo la distancia del frente de Pareto generado

por el algoritmo propuesto hacia el frente de Pareto exacto, y maximizar la expansión de soluciones encontradas. Las métricas de desempeño pueden ser clasificadas en tres categorías: minimizar la distancia del frente de Pareto producido por el algoritmo con respecto al frente verdadero (suponiendo se conoce), maximizar la distribución de soluciones obtenidas, de manera que sea posible tener una distribución de vectores tan uniforme como sea posible y maximizar la cantidad de elementos del conjunto de óptimos de Pareto generados (Zitzler, 2000).

Dentro de la literatura se encuentran algunas de las siguientes métricas para evaluar el desempeño algorítmico.

#### **2.4.1 Distancia generacional**

Esta métrica fue presentada por Van Veldhuizen y Lamont (Van, 1998) sirve para medir la distancia de las soluciones que están en el conjunto de soluciones no dominadas encontradas hasta ahora de los del conjunto óptimo de Pareto y se define como:

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n}$$

Donde  $n$  es el número de soluciones en el conjunto de soluciones no dominadas encontrado hasta el momento y  $d_i$  es la distancia euclidiana (medido en el espacio objetivo) entre cada una de estas soluciones y el miembro más cercano del conjunto de Pareto óptimo. Está claro que un valor de  $GD = 0$  indica que todos los elementos generados están en el conjunto óptimo de Pareto. Con el fin de obtener resultados fiables, los conjuntos de soluciones no dominadas se normalizan antes de calcular la medida de distancia.

#### **2.4.2 Distancia Generacional Invertida (IGD)**

Determina el promedio de las distancias entre cada solución del frente de Pareto verdadero a la solución más cercana del frente de Pareto encontrado. Este

indicador mide la proximidad que existe entre el frente de Pareto verdadero ( $FPv$ ) y el frente de Pareto encontrado. Se formula de la siguiente manera (Jiang, 2014):

$$IGD \triangleq \frac{(\sum_{i=1}^n d_i^p)^{\frac{1}{p}}}{n}$$

Donde  $n$  es el número de soluciones en el Frente de Pareto verdadero,  $p$  es 2, y  $d_i$  es la distancia Euclidiana en el espacio de los objetivos entre cada vector de frente de Pareto verdadero y el punto más cercano del frente de Pareto encontrado. Si el resultado es 0 indica que ambos frentes son iguales; cualquier otro valor indica una divergencia entre ambos frentes; por ello es deseable obtener valores bajos de IGD.

### 2.4.3 Spread

La métrica Spread (Deb, 2002) es una métrica de diversidad que mide el grado de propagación logrado entre las soluciones obtenidas. Esta métrica se define de la siguiente manera:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N - 1)\bar{d}}$$

Donde  $d_i$  es la distancia euclidiana entre  $N$  soluciones consecutivas,  $\bar{d}$  es la media de estas distancias, y  $d_f$  y  $d_l$  son las distancias euclidianas a las soluciones extremas (delimitador) del frente de Pareto exacta en el espacio objetivo. Esta métrica toma un valor de cero para una distribución ideal, señalando una extensión perfecta de las soluciones en el frente de Pareto. Se aplica esta métrica después de una normalización de los valores de la función objetivo.

### 2.4.4 Tasa de errores

Veldhuizen introdujo esta métrica la cual cuenta el número de soluciones óptimas en el frente de Pareto encontrado las cuales pertenecen al frente de Pareto

verdadero y lo divide por el número de soluciones encontradas. La formulación de este indicador es el siguiente (Jiang, 2014):

$$ER = \frac{\sum_{i=1}^n e_i}{n}$$

El valor más bajo de esta medida muestra la mejor aproximación de las verdaderas soluciones óptimas de Pareto.

### 2.4.5 Hipervolumen (HV)

Hipervolumen es un indicador combinado de convergencia y diversidad que calcula el volumen (en el espacio objetivo) cubierto por los miembros de un conjunto no dominado de soluciones  $Q$ . Matemáticamente, para cada solución  $i \in Q$ , un hipercubo  $v_i$  se construye con un punto de referencia  $W$  y la solución  $i$  como las esquinas diagonales de la hipercubo. El punto de referencia, simplemente se puede encontrar mediante la construcción de un vector de los peores valores de la función objetivo. A partir de entonces, se encuentra la unión de todos los hipercubos y su hipervolumen (HV) se calcula como sigue (Zitzler, 1999):

$$HV = \text{volume} \left( \bigcup_{i=1}^{|Q|} (v_i) \right)$$

Donde:

$Q$  es un conjunto de soluciones no dominadas,

$i$  es una solución de  $Q$ ,

$v_i$  es un hipercubo de la solución  $i$ .

La Figura 2 muestra un ejemplo de dos objetivos a minimizar donde  $Q = \{A, B, C\}$  es la región acotada por la línea discontinua. Para cada solución  $i \in Q$ , se construye un hipercubo  $v_i$  utilizando un punto de referencia  $W$  (por ejemplo, puede formarse tomando el peor valor de cada objetivo de las soluciones) y la solución  $i$  como las esquinas de la diagonal del hipercubo. Los algoritmos que alcanzan mayores

valores para HV son mejores. Así, HV se calcula como el volumen de la unión de todos los hipercubos.

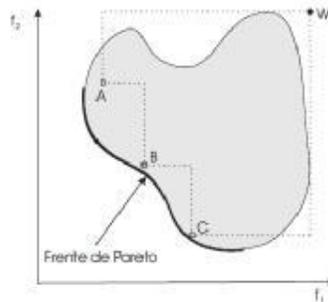


Figura 2 Hipervolumen cubierto por las soluciones no dominadas a, b, c

#### 2.4.6 Indicador de Dominancia de Pareto (NR)

Mide el radio de soluciones no dominadas proporcionadas por una frontera aproximada particular con respecto a soluciones no-dominadas proporcionadas colectivamente por todos los frentes aproximados (Deb, 2002).

#### 2.4.7 Cobertura de conjuntos

Esta métrica compara dos conjuntos de soluciones en términos de Pareto dominancia relativa, es decir, presenta resultados del porcentaje de soluciones de un conjunto que son dominadas por alguna solución de otro conjunto, se define de la siguiente manera (Jiang, 2014):

$$C(A, B) = \frac{|\{b \in B | \exists a \in A: a \preceq b\}|}{|B|}$$

$C(A, B)$  Proporción de soluciones en  $B$ , débilmente dominadas por soluciones en  $A$ .

$C(A, B) = 1$  Todos los miembros de  $B$  son dominados débilmente por  $A$ .

$C(A,B) = 0$  Ningún miembro de  $B$  es débilmente dominados por  $A$ .

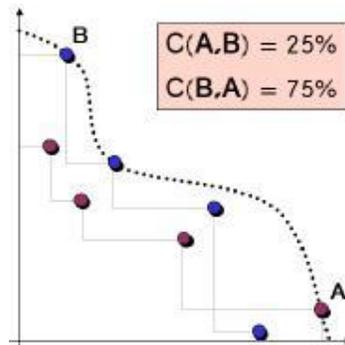


Figura 3. Representación de Cobertura de Conjuntos.

#### 2.4.8 Distribución Uniforme (UD)

Cuantifica el grado de aproximación a una distribución uniforme de un frente de aproximación bajo un parámetro pre-definido.

#### 2.4.9 Espaciado (S)

Mide que tan uniformemente las soluciones evolucionadas se distribuyen.

#### 2.4.10 Máxima amplitud (MS)

Mide que tan bien la frontera de Pareto verdadera es cubierta por el conjunto aproximado.

### 2.4.11 Análisis de métricas de desempeño

La Tabla 1 presenta características importantes de métricas de desempeño muy aceptadas en la literatura. La mayoría de las características consideradas fueron tomadas de (Bechikh, 2013) y se describen a continuación.

**TABLA 1. CARACTERÍSTICAS DE INDICADORES DE DESEMPEÑO**

Indicador de Desempeño	Unario	Binario	Convergencia	Diversidad	Mejor valor	Mejor medición
<b>HV</b>	X		X	X	1	Frente de Pareto convexo
<b>UD</b>	X			X	1	Frente de Pareto continuo
<b>S</b>	X			X	0	Hasta 2 objetivos
<b>MS</b>	X			X	0	Hasta 2 objetivos
<b><math>\Delta</math></b>	X			X	0	Desconocida
<b>GD</b>	X		X		0	Desconocida
<b>IGD</b>	X		X		0	Desconocida
<b>SC</b>		X	X		1	Desconocida

a) Unario. Indica si es un indicador de desempeño unario (ej. Medida de desempeño que asigna un único valor a cada conjunto de soluciones no-dominadas).

b) Binario. Indica si es un indicador de desempeño binario (ej. Medida de desempeño que asigna un único valor a un par de conjuntos de soluciones no-dominadas).

c) Convergencia. Indica que el indicador de desempeño asigna un único valor que corresponde a la convergencia del conjunto de soluciones no-dominadas.

d) Diversidad. Indica que el indicador de desempeño asigna un único valor que corresponde a la divergencia del conjunto de soluciones no-dominadas.

- e) Mejor valor. El mejor valor que puede ser obtenido del indicador de desempeño. Por ejemplo, a mayor valor de HV mejor es el resultado, siendo el máximo valor 1.
- f) Mejor medición. Condición para que la métrica no de resultados erróneos.

Ninguna métrica por si sola puede medir de manera satisfactoria el desempeño de un MOEA. Cada métrica puede proporcionar una cuantificación del desempeño específico, pero incompleto y solo puede ser efectiva bajo ciertas condiciones. Por ejemplo, UD trabaja de manera pobre cuando el frente de Pareto es discontinuo, mientras que indicador de hipervolumen puede dar resultados erróneos si el frente de Pareto es no-convexo.

**TABLA 2. MÉTRICAS PARA CUANTIFICAR EL DESEMPEÑO DE MOEAS**

Autores	Artículo	Métricas utilizadas en experimentos
(Bechikh, 2013)	Incorporating Decision Maker's Preference Information in Evolutionary Multi-objective Optimization	Spread (de la ROI), GD
(Xie, 2013)	A Multi-Algorithm Balancing Convergence and Diversity for Multi-Objective Optimization	HV, IGD, and Spacing
(Yen, 2014)	Performance Metric Ensemble for Multiobjective Evolutionary Algorithms	NR, IGD, MS, y HV
(Baños, 2006)	Meta-heurísticas Híbridadas para Optimización Mono-objetivo y Multi-objetivo. Paralelización y Aplicaciones	CS, HV

La Tabla 2 presenta algunos trabajos, se puede observar que existen diferencias en el conjunto de métricas utilizado. Esto pone de manifiesto la necesidad de estudiar bajo qué condiciones cada métrica tiene su mejor comportamiento. La práctica común es explotar varias métricas para determinar cuál combinación de métricas ofrece una mejor opción, sin embargo, este proceso tiene un costo computacional alto (Yen, 2014). En esta tesis también es de interés el estudio de las métricas para analizar la relación entre la diversidad de las soluciones de las poblaciones por las que el algoritmo transita y su efecto en el desempeño parcial y final.

# Capítulo 3

## III. ESTADO DEL ARTE

---

En este capítulo se presenta una breve revisión de los trabajos relacionados que dan solución al Problema Path Dissimilarity los cuales han mostrado resultados sobresalientes. También se presentan los trabajos relacionados que emplean métodos de caracterización.

### **3.1 OBTENCIÓN DE EXPLICACIONES DEL DESEMPEÑO ALGORÍTMICO**

El problema Path Dissimilarity ha sido estudiado con distintas formulaciones, destacando los trabajos de Martí (Duarte,2007)

Martí, González-Velarde, y Duarte (2009) (Duarte, 2009) dieron solución a PDP mediante la heurística GRASP. Primero, el algoritmo genera  $k$ -rutas más cortas de acuerdo a valores de riesgo y las ordena para después eliminar las rutas duplicadas. Después, en la búsqueda local se realizan intercambios eliminando un camino y agregando uno nuevo con la finalidad de mejorar la solución actual. La evaluación y mejora de los objetivos se realiza de manera secuencial: primero minimiza el valor del primer objetivo sin considerar el segundo objetivo. En la segunda etapa se trata de mejorar el segundo objetivo, independientemente del primer objetivo. Esta operación se realiza hasta que no exista una mejora o hasta que se alcance el número máximo de iteraciones.

(Martí, 2012) extienden su enfoque de solución agregando a la heurística GRASP la propiedad de ruta enlazada (path relinking). Además, incluyeron otros operadores para la generación de soluciones: generación de soluciones totalmente aleatorias,

generación de soluciones guiadas por los objetivos de manera alterna sin que el valor de un objetivo influya sobre el otro y la generación de soluciones basada en la combinación de las dos primeras.

Renan (2016) (Renan, 2016) emplea NSGA-II para resolver el problema multi-objetivo VRP con respuesta de demanda de transporte de (VRPDRT), el cual es un problema de rutas en donde un tipo de transporte permite a los clientes tomar a su destino como un taxi con el fin de reducir los costos de operación y para satisfacer las necesidades del cliente.

## **3.2 OBTENCIÓN DE EXPLICACIONES DEL DESEMPEÑO ALGORÍTMICO**

Generalmente, la explicación de una conducta algorítmica está basada en la identificación de medidas de caracterización, por lo que esta sección inicia con la revisión de este concepto, seguido del análisis individual de trabajos.

### **3.2.1 Medidas de caracterización**

En cualquier área siempre se desea alcanzar el éxito, y para medir este éxito deben formularse elementos que sirvan para definir si algo fue bueno o no, o si cumplió con lo esperado o no. Por lo que es importante que se establezcan metas medibles e indicadores de desempeño. Sin una base concreta para medir el éxito, la evaluación de cualquier trabajo se vuelve imprecisa y subjetiva, para ello existen los indicadores y métricas (Fiksel, 1998).

Un indicador de desempeño es un atributo específico que muestra, da indicios o señales de algún aspecto a evaluar. Es una unidad de medida que permite el seguimiento y la evaluación periódica de los resultados obtenidos en la ejecución de un programa, proyecto o actividad por lo que cada indicador debe estar asociado con al menos una métrica (Fiksel, 1998).

Una métrica es “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado” (Ejiogu, 1991). Es la aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo de un producto para suministrar información relevante a tiempo, para mejorar el proceso y el producto (Serrano, 2002).

Las métricas deben ser verificables, objetivas y significativas. Existen dos grandes categorías de métricas: las primeras son métricas cuantitativas que se basan en datos empíricos y caracterizan el rendimiento numéricamente. La segunda categoría es métricas cualitativas que dependen de distinciones semánticas basadas en la observación y el juicio (Fiksel, 1998).

Una de las metas de usar métricas de caracterización es aprender y entender la relación entre características de la instancia y el desempeño de algoritmo. En la literatura especializada, se han propuesto métricas (independientes o dependientes del problema) las cuales han sido relacionadas con cada etapa del proceso de solución de algunos problemas de optimización, mediante modelos de regresión (Weber, 2012), metodologías para identificar el punto de transición de problema fácil a difícil (phase transition) [(Boettcher,2004)(Martín,2001)], metodologías para analizar la superficie de aptitud (Knowles,2006) y, en forma más general, mediante métodos de descubrimiento de conocimiento del área de aprendizaje automático. La revisión de la literatura revela la necesidad de formular métricas suficientes, relevantes y complementarias para problemas afines (Smith-Miles, 2012).

Existen muchas extensiones del análisis de superficie para problemas multi-objetivo [(Deb, 1999), (Knowles, 2002)]. Sin embargo, pesar del amplio estudio, aún es poco entendida la relación entre la dificultad de un problema y la superficie de aptitud de muchos problemas de optimización combinatoria multi-objetivo y mono-objetivo (Bierwirth, 2004). El uso complementario de métricas podría ser una alternativa más reveladora.

En algunos trabajos se ha utilizado el término índice para nombrar a las medidas de caracterización del proceso de solución algorítmico, para estandarizar términos y acorde con las definiciones previas, en la presente investigación se denominarán métricas.

#### **3.2.1.1 Problema**

En la entrada, se deben tomar en cuenta las métricas que permitan dar a entender las características de la instancia como es su estructura y su dificultad, por ejemplo, el tamaño de la instancia, la estructura del espacio de búsqueda y de las soluciones óptimas, también se debe considerar aquellas métricas que proporcionen información a partir de los parámetros que integran el problema.

Es importante considerar medidas estadísticas que se apliquen a los valores que toman los parámetros de la instancia, para desde ese punto tener cierto conocimiento para comprender el comportamiento del algoritmo y su resultado final.

#### **3.2.1.2 Algoritmo**

Durante el proceso las métricas que se deben seleccionar son aquellas que caractericen el comportamiento del algoritmo en tiempo de ejecución. Algunas de las características que es importante medir son las características de la función objetivo, las características del espacio de búsqueda además de medidas estadísticas aplicadas a los resultados parciales de la búsqueda.

#### **3.2.1.3 Resultado final**

En la salida, se deben considerar métricas de desempeño algorítmico, por ejemplo: el tiempo de solución y calidad de solución. Además, se deben aplicar medidas estadísticas sobre los resultados finales de la búsqueda. Por ejemplo, sobre la distribución del tiempo de ejecución.

### **3.2.2 Revisión de trabajos de la literatura**

En la literatura hay gran cantidad de artículos donde relacionan las características de algunas clases de instancias de problemas específicos con el desempeño de varios algoritmos. Muestra de ello son los siguientes:

Lemeire y Dirx (Lemeire, 2004), mediante la construcción de un modelo causal analizan el desempeño de un algoritmo paralelo de multiplicación de matrices, en donde descubren las principales causas que originan varias anomalías observadas en el proceso de comunicación de distintas corridas del algoritmo.

Hoos y Stützle (Hoos, 2004) mediante un algoritmo de búsqueda local iterativa dan solución a instancias del problema MAX-SAT analizando el comportamiento y desempeño de un algoritmo. Las trayectorias de búsqueda trazadas por el algoritmo para resolver instancias del problema son caracterizadas por diferentes métricas de correlación.

Chevalier (Chevalier, 2006) realiza un estudio experimental, donde considera diferentes configuraciones de los parámetros de un algoritmo genético para resolver el problema de diagnosis de fallas múltiples, obteniendo un balance correcto entre la exploración y la explotación del espacio de búsqueda para obtener mejores individuos.

En el trabajo de Pérez (Pérez, 2007) se presenta una metodología la cual realiza un estudio sistemático de los elementos que intervienen en el proceso algorítmico asociado a una metaheurística, búsqueda tabú. Con esta metodología se realiza un estudio experimental con el cual se identifican aspectos que influyen en el desempeño algorítmico.

Landero (Landero, 2008) presenta una metodología similar a la propuesta en el trabajo de Pérez, sin embargo, en lugar de partir de un análisis experimental para

identificar variables e indicadores del proceso algorítmico, se obtienen métricas que caracterizan las instancias del problema y la estructura interna del algoritmo.

Quiroz (Quiroz, 2014) combinó métodos de análisis exploratorio de datos y de análisis causal en cuatro etapas. En la etapa de caracterización se identifican factores que influyen en el desempeño y se cuantifican mediante índices.

### **3.2.3 Análisis comparativo de trabajos relacionados**

Algunos trabajos relacionados son presentados en la Tabla 3. La primera columna son las referencias de los trabajos considerados. La segunda columna indica, para cada trabajo, si se realizó un análisis experimental inicial con el objetivo de identificar variables y métricas de caracterización del proceso de solución algorítmico (AEI, Análisis Experimental Inicial).

La tercera y cuarta columnas indican si el análisis del algoritmo se realiza de manera conjunta con la información de instancias del problema; este análisis puede realizarse sobre la descripción de la instancia del problema (MDP, Métricas de Descripción del Problema) o de una muestra del espacio de soluciones de la instancia del problema (MEP, Métricas del Espacio del Problema).

Las columnas de la quinta a la séptima indican si la información del algoritmo es contemplada en el análisis; dicha información puede ser obtenida del comportamiento del algoritmo en relación a sus principales parámetros de control (MCA, Métricas de Comportamiento del Algoritmo), trayectoria de búsqueda (MTB, Métricas de Trayectoria de Búsqueda) y estructura interna del algoritmo (MEI, Métricas de Estructura Interna). La octava columna indica si los trabajos relacionados han realizado explicaciones formales del desempeño de algoritmos. La columna final muestra si los trabajos se realizaron como problemas multi-objetivo (Quiroz, 2014).

El análisis de trabajos relacionados revela que algunos de ellos [(Hoos, 2004), (Konak, 2005), (Chevalier, 2006)] proveen explicaciones a través de un análisis tabular de resultados experimentales del comportamiento de un algoritmo sobre un conjunto de instancias de un problema. Otros trabajos [(Lemeire, 2004), (Pérez, 2007), (Pérez J., 2008)] llevan a cabo un análisis causal usando algoritmos de aprendizaje de estructura y de parámetros, y proveen un modelo formal del comportamiento de un algoritmo.

El trabajo de Pérez (Pérez, 2007) revela formalmente porqué un algoritmo se desempeña mejor al resolver un subconjunto de instancias, aunque, su explicación es limitada, pues no se incluyen todos los tipos de índices del problema y del algoritmo.

La metodología propuesta por Landero (Landero, 2008) permite relacionar índices influyentes de la dominación de un algoritmo y la estructura interna del mismo, para proveer explicaciones a través del análisis causal, además de validar y aplicar el modelo causal obtenido.

**TABLA 3. TRABAJOS RELACIONADOS CON LA CARACTERIZACION DEL DESEMPEÑO DE ALGORITMOS**

Trabajo	Identificación inicial de variables AEI	Información analizada					Explicaciones formales	Problemas multi-objetivo
		Métricas de las instancias del problema		Métricas del algoritmo				
		MDP	MEP	MCA	MTB	MEI		
(Hoos, 2004)		x			x			
(Konak, 2005)						x		
(Lemeire, 2004)				x			x	
(Chevalier, 2006)				x				
(Pérez, 2007)	x			x			x	
(Landero, 2008)			x	x			x	
(Quiroz, 2014)	x	x	x	x	x	x	x	
En este trabajo	x	x	x	x	x	x	x	

Quiroz (Quiroz, 2014) realizó un análisis con enfoque experimental para un estudio integral del proceso de optimización con la finalidad de identificar relaciones entre los factores que afectan al desempeño algorítmico. Combinó métodos de análisis

exploratorio de datos y de análisis causal en cuatro etapas. Para evaluar el trabajo realizado analizó el proceso de optimización del problema de empaqueo de contenedores. Además, identificó propiedades que definen la estructura de una instancia de BPP. Generó nuevos conjuntos de instancias difíciles.

El trabajo presente realizará una adaptación de un algoritmo que da solución a un problema de agrupación mono-objetivo de agrupación para que este algoritmo sea capaz de resolver un problema muy relacionado, pero con un enfoque multi-objetivo. Esto se realizará mediante el uso de métricas seleccionadas de la literatura las cuales permitan caracterizar al algoritmo durante todo el proceso de solución del problema, es decir, entrada, proceso algorítmico y estructura de las soluciones obtenidas, de manera que se alcance el mejor desempeño. El problema multi-objetivo con el que se trabajará deberá ser similar al problema de empaqueo de contenedores.

# Capítulo 4

## IV. ADAPTACIÓN DEL ALGORITMO GGA-CGT

---

En este capítulo, cada método dentro del Algoritmo Genético con Transmisión Controlada de Genes (Quiroz, 2014) es estudiado a fin de conocer las estrategias utilizadas y realizar la adaptación del mismo con el objetivo de dar solución a un problema multi-objetivo.

### 4.1 ESTRUCTURA ORIGINAL DEL ALGORITMO

En el Algoritmo 1 se muestra el esquema general del algoritmo GGA-CGT en donde se aprecia la estructura de acuerdo a un algoritmo genético. El proceso inicia con la generación de una población  $P$  de individuos generados con la heurística FF- $\tilde{n}$  (Línea 1).

---

#### Algoritmo 1. GGA-CGT

---

```
1  Generar una población inicial  $P$  con FF –  $\tilde{n}$ ;  
2  while generación <  $max\_gen$  and Size( $mejor\_solucion$ ) >  $L_2$   
3      Seleccionar  $n_c$  individuos para cruzar con Controlled_Selection;  
4      Aplicar Gene_Level_Crossover + FFD a  $n_c$  individuos seleccionados;  
5      Aplicar Controlled_Replacement para introducir descendencia;  
6      Seleccionar  $n_m$  individuos y clonar soluciones élite mediante Controlled_Selection;  
7      Aplicar Adaptive_Mutation + RP a los mejores individuos de  $n_m$ ;  
8      Aplicar Controlled_Replacement para introducir clones;  
9      Actualización de global_best_solution;  
10 end;  
Fin GGA-CGT
```

---

En la primer fase (líneas 3-5)  $n_c$  individuos son seleccionados por el método *Controlled\_Selection*, y se les aplica el operador *Gene\_Level\_Crossover+FFD*, después para generar a los hijos se usa el operador *Controlled\_Replacement*.

En la fase siguiente (líneas 6-8), individuos élite son clonados de acuerdo al valor de *life\_span* y los mejores  $n_m$  individuos son seleccionados para ser modificados genéticamente mediante el operador Adaptive\_Mutation+RP, después, los individuos que son clon son introducidos a la población mediante Controlled\_Replacement. En cada generación la mejor solución global se actualiza como se indica en la línea 9. Y este proceso se repite tantas veces se le indique en *max\_gen*.

## 4.2 REDISEÑO DE LA ESTRUCTURA DEL ALGORITMO

Como ya se ha mencionado, un algoritmo genético (GA) está inspirado en el proceso genético de los organismos vivos, es una estrategia que emula el proceso evolutivo natural. Un GA crea soluciones factibles (población) para un problema, las cuales haciendo uso de la evolución se van combinando (mutación) y alterando (cruza) para crear soluciones más fuertes (Quiroz, 2014). En optimización mono-objetivo es una función de aptitud la que determina qué solución prevalece contra el resto, pero al tratarse de varios objetivos una poderosa alternativa es usar la característica de frentes de dominancia, para encontrar qué conjunto de soluciones son las mejores, estas se encuentran en el frente cero mejor conocido como Frente de Pareto.

Se ha denominado M-GGA-CGT a la adaptación multiobjetivo producida a partir de GGA-CGT. Esta extensión conserva las estrategias originales del algoritmo, excepto la Línea 9, la cual posibilita la solución al problema multi-objetivo Path Dissimilarity. Líneas abajo se detallan los cambios realizados en cada módulo, los cuales se han nombrado en español con fines didácticos.

El algoritmo inicia con la generación de una población inicial  $P$  de individuos generados con la heurística  $FF-\tilde{n}$  (Línea 1). En la primera fase (líneas 3-5)  $n_c$  individuos son seleccionados por el método *SeleccionControladaParaCruza*, y se les aplica el operador *CruzaNivelGen+FFD* para generar a los hijos de éstos.

---

**Algoritmo 2. M-GGA-CGT**

---

```
1  $P \leftarrow PoblacionInicial + FF - \tilde{n}$ .
2 while (generación <  $max\_gen$ )
3      $n_c \leftarrow SeleccionControladaParaCruza(P)$ 
4      $H \leftarrow CruzaNivelGen + FFD(n_c)$ 
5      $P \leftarrow ReemplazoControladoDeCruza(H)$ 
6      $n_m \leftarrow SeleccionControladaParaMutacion(P)$ 
7      $n_m \leftarrow MutacionAdaptativa + RP(n_m)$ 
8      $P \leftarrow ReemplazoControladoDeMutacion(n_m)$ 
9      $F_0 \leftarrow ActualizacionFrente(P)$ 
10 end
Fin M-GGA-CGT
```

---

Mediante el operador *ReemplazoControladoDeCruza* se reemplazan varios elementos de la población por los hijos generados. En la fase siguiente (líneas 6-8), los mejores individuos son clonados y los mejores  $n_m$  individuos son seleccionados para ser modificados genéticamente mediante el operador *MutacionAdaptativa+RP*, después, los individuos mutados son introducidos a la población mediante el operador *ReemplazoControladoDeMutacion*. En cada generación se actualiza el Frente de Pareto (línea 9). Estos procesos se repiten tantas veces se le indique en  $max\_gen$ .

## 4.2.1 Componentes del algoritmo M-GGA-GCT

### 4.2.1.1 Población inicial y heurística FF- $\tilde{n}$

Para generar la población inicial se hace uso de la heurística FF- $\tilde{n}$ , esta heurística genera una solución con diferentes rutas y cada ruta es considerada como un grupo. Esta heurística se describe de la siguiente manera: dado un conjunto  $N$  de  $n$  vértices, se genera el subconjunto  $\tilde{N}$  de  $\tilde{n}$  vértices donde  $\tilde{n}_i$  tiene una distancia respecto al origen y dicha distancia es 50% o más grande que el límite de distancia asignado para cada ruta. Primero, los  $\tilde{n}$  vértices más grandes de  $\tilde{N}$  son acomodados en rutas diferentes; a continuación, cada uno de los vértices restantes se colocan uno a uno en la primera ruta que tiene la capacidad suficiente para alojar

al vértice; si ninguna de las rutas puede almacenarlo, el vértice se coloca en una nueva ruta y así continúa hasta que todos los vértices estén dentro de una ruta.

#### 4.2.1.2 Selección controlada para Cruza

Este operador se encarga de seleccionar qué elementos serán cruzados. Se obtiene un conjunto  $n_c$  de parejas  $G$  y  $R$ , donde  $G$  pertenece al conjunto de soluciones buenas y  $R$  pertenece al conjunto de soluciones aleatorias. Posteriormente estos elementos serán combinados en pareja  $G_i$  con  $R_i$ . El subconjunto  $G$  contiene  $n_c/2$  elementos tomados de la población de forma aleatoria y el subconjunto  $R$  contiene  $n_c/2$  elementos tomados dentro de las mejores soluciones.

#### 4.2.1.3 Cruza a nivel gen + FFD

Dada dos soluciones  $P_1$  y  $P_2$ , se cruzan dos veces para generar dos hijos:  $P_1$  con  $P_2$  y  $P_2$  con  $P_1$ . Este operador considera los grupos en orden descendente de su llenado, para aumentar la probabilidad de que los mejores grupos serán heredados por los hijos. Comenzando en el mejor grupo más lleno, los grupos individuales de ambos padres se comparan en paralelo, descartando los grupos en donde se repiten los elementos; al final aquellos elementos que hayan quedado fuera son reinsertados con la heurística muy conocida FFD (Quiroz, 2014).

El proceso de cruce se describe de la siguiente manera:

Capacidad ocupada	10	9	8	8	5
Elementos dentro	3,5	8,2	0,6	1,7	4
Peso de los elemento	8,2	2,7	6,2	3,5	5

Primer padre

Capacidad ocupada	10	8	8	7	7
Elementos dentro	1,2	3	0,8	4,6	7,5
Peso de los elementos	3,7	8	6,2	4,2	5,2

Segundo padre

Dado:

- una instancia P,
- contenedores con capacidad de 10,
- un conjunto N con 9 elementos,
- el peso de los elementos (6,3,7,8,5,2,2,5,2).

Se ordenan de acuerdo al peso del contenedor para generar al hijo

Capacidad ocupada	10	10	9	8	8	8	8	7	7	5
Elementos	3,5	1,2	8,2	3	0,6	0,8	1,7	4,6	7,5	4

Se seleccionan los contenedores llenos o más llenos de manera que no se repitan los elementos, quedando libres los elementos 7 y 8 sin ubicación.

Capacidad ocupada	10	10	8	5
Elementos	3,5	1,2	0,6	4

Para asignar los elementos que quedaron libres se usa la heurística FFD, la cual consiste en buscar en qué contenedor aún existe espacio de manera que el objeto pueda añadirse. Quedando de la siguiente manera:

Capacidad ocupada	10	10	10	10
Elementos	3,5	1,2	0,6,8	4,7

#### 4.2.1.4 Reemplazo controlado de Cruza

El operador de cruza ha generado un conjunto  $C$  de los hijos de  $n_c$ . Los primeros  $n_c/2$  hijos se introducen a  $P$  y reemplaza a los individuos en el conjunto de los padres aleatorios  $R$ . Los  $n_c/2$  hijos restantes se introducen en  $P \setminus (G \cup R)$  con las siguientes condiciones: (a) reemplazar individuos con aptitud duplicada; (b) si hay hijos que no han sido insertados en  $P$ , entonces introducirlos, en sustitución de las peores soluciones.

#### 4.2.1.5 Selección controlada para mutación

Se seleccionan  $n_m$  elementos de  $P$  los cuales pertenecen al grupo de las mejores soluciones y cumplen con la condición de que la generación en la que fueron creados sea menor al valor de *life span*. Éste parámetro es constante e indica el límite para que una solución pueda ser mutada, esto en base al número de generación en la que dicha solución fue creada.

#### 4.2.1.6 Mutación adaptativa + RP

El proceso de mutación funciona a nivel del gen (grupo o ruta), el cual propicia la transmisión de los mejores genes en el cromosoma, sin perder los mejores patrones individuales (aquellos grupos completos), este operador considera los grupos en el orden de su llenado descendente, elimina aquellos grupos más vacíos y vuelve a insertar los vértices con la heurística de reordenamiento RP, la cual se describe líneas abajo. Para determinar cuántos grupos serán eliminados de la solución se hace uso de la siguiente función:

$$n_b = \lceil i \cdot \varepsilon \cdot \rho_\varepsilon \rceil$$

$$\varepsilon = \frac{2 - (i/m)}{i^{1/k}}$$

$$\rho_\varepsilon = 1 - \text{Uniform}(0, \frac{1}{i^{1/k}})$$

Donde:

$m$  es la cantidad de contenedores,

$i$  es la cantidad de contenedores incompletos,

$k$  es un parámetro que define la proporción de cambio de  $\varepsilon$  y  $\rho_\varepsilon$  con respecto a  $i$ .

El proceso de cruce se describe de la siguiente manera:

Dado:

- una instancia  $P$ ,
- contenedores con capacidad de 10,
- un conjunto  $N$  con 9 elementos,
- el peso de los elementos (6,3,7,8,5,2,2,5,2).

Capacidad ocupada	10	10	9	8	8	4
Elementos dentro	3,5	1,7	8,2	0,6	9	4

Solución

Se tiene en la solución  $m = 6$  número de contenedores, el número de contenedores incompletos  $i = 4$  y la proporción de cambio  $k = 3$ . Mediante las ecuación de mutación se obtiene  $\varepsilon = 0,83$  y la probabilidad de eliminación  $\rho_\varepsilon = 0,5$ . La cantidad de contenedores a eliminar es  $n_b = 2$ , lo que indica que al menos 2 contenedores serán eliminados de la solución. Quedando libres los elementos 9 y 4.

Capacidad ocupada	10	10	9	8
Elementos dentro	3,5	1,7	8,2	0,6

Para ubicar los elementos que han quedado libres se usa un método llamado RP, el cual consiste en comparar cada elemento de cada contenedor con los elementos

libres con la finalidad de mejorar el espacio ocupado del contenedor, después que se han realizado todas las comparaciones a los elementos que queden libres se acomodan usando la heurística FF haciendo uso de un nuevo contenedor para almacenar al elemento restante.

Capacidad ocupada	10	10	9	10	10
Elementos dentro	3,5	1,7	8,2	9,6	4,0

Solución

#### 4.2.1.7 Reemplazo controlado de mutación

Cuando se aplica el operador de mutación, algunos de los mejores individuos son clonados para preservar las mejores soluciones. Cada clon se puede introducir en la población de dos maneras: (a) si hay soluciones con la aptitud duplicada, a continuación, el clon sustituirá a una de ellos; y (b) si la primera alternativa no es posible, entonces el clon reemplazará a la peor solución.

#### 4.2.1.8 Actualización de los Frentes

En cada iteración se obtienen los Frentes ordenando la población de acuerdo al grado de dominancia de manera tal que las mejores soluciones quedan al inicio de la población [5].

### 4.3 CONFIGURACIÓN DE PARÁMETROS DEL ALGORITMO M-GGA-GCT

Se ha implementado una metodología con enfoque experimental que evalúa el impacto de los parámetros en el rendimiento de un algoritmo y, por este medio, reduce el número de experimentos necesarios. Este enfoque combina las Covering Arrays y el análisis exploratorio de datos. El trabajo propuesto tiene las siguientes características:

- a) El espacio de parámetros se explora a través de muestras de tamaño mínimo, pero suficiente para inferir las relaciones entre cada par de variables, que cubre a través de las matrices.
- b) Los efectos de los parámetros sobre el rendimiento final se aproximan a medida que se realizan las técnicas de análisis exploratorio de datos.
- c) Los valores de las pruebas y sus áreas de distribución se actualizan dinámicamente
- d) El ajuste de parámetros es fuera de línea.

### 4.3.1 Metodología propuesta para la configuración de parámetros

Este trabajo se describe de la siguiente manera (Figura 4):

Dados:

- Un conjunto de instancias  $I$
- Un algoritmo  $A$  para resolver  $I$
- Un conjunto  $P$  de  $k$  parámetros
- El rango de valores  $[p_i^{\min}, p_i^{\max}]$  para cada parámetro  $i$ .
- El tamaño de rango mínimo requerido  $R_i^{\min}$  para cada parámetro  $i$ .
- Los valores  $v$  y  $t$  para el tamaño del alfabeto y la cobertura del CA.

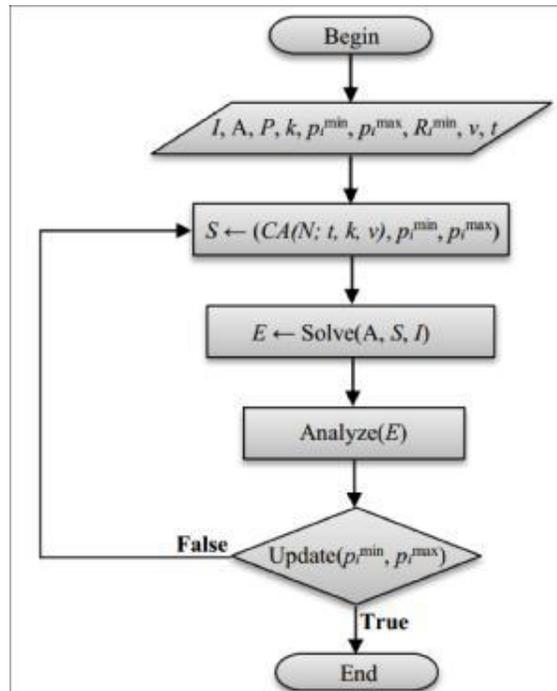
El proceso metodológico consiste en cuatro pasos:

1.-  $S$ . Teniendo todos los valores de entrada se realizará el cálculo de los valores del conjunto de configuraciones  $S$  los cuales se obtienen mediante la siguiente ecuación:

$$s_{ji} = p_i^{\min} + \frac{(CA_{ji} + 1)(p_i^{\max} - p_i^{\min})}{v + 1} \quad \forall j \in \{0, \dots, N-1\}, \forall i \in \{0, \dots, k-1\}$$

El conjunto de configuraciones  $S$  es una matriz de tamaño  $N \times k$ , calculado de acuerdo a la ecuación 1, donde  $p_i^{\min}$  y  $p_i^{\max}$  son los valores máximos y mínimo para el parámetro  $i$ ,  $CA_{ji}$  es el elemento en la fila  $j$  y la columna  $i$  de la matriz de  $CA(N; t, k, v)$  y  $v$  es el número de valores para cada parámetro.

Fig 4. Metodología Propuesta para la configuración de parámetros



**2.- Solve.** Este paso consiste en probar cada configuración  $S_i$  generada en el paso anterior, alimentando al algoritmo  $A$  por cada instancia  $I$ . Como en esta experimentación se desea que la instancia  $I$  alcance el valor óptimo, se deberá realizar un acumulado de cuántas instancias han alcanzado el valor óptimo con cada configuración  $S_i$  en cada ejecución.

Teniendo estos valores, se guardarán en un archivo  $E$ , el cual será usado para que se le apliquen varios métodos estadísticos en el siguiente paso.

**3.- Analyze(E).** La finalidad de este paso es encontrar aquel parámetro que propicie que el algoritmo tenga un mejor desempeño. Cuando se haya encontrado dicho parámetro éste se utilizará como base, y se reducirá el rango que este pueda abarcar, para que en una siguiente iteración se trabaje con un espacio más pequeño y que a la vez permita obtener mejores resultados. El proceso para encontrar a este

parámetro es el siguiente y se basa en los datos obtenidos en el archivo  $E$  generado previamente:

Se calcula el Coeficiente de correlación a cada conjunto de valores de cada parámetro con respecto al número de instancias que alcanzaron el valor óptimo. Al valor final de cada parámetro se utiliza con valor absoluto.

Se toman los primeros 6 valores de configuración de cada parámetro, y se obtiene el promedio de cuántas instancias haciendo uso de ese valor de configuración alcanzaron el valor óptimo. Se obtendría un conjunto de 6 elementos para cada parámetro.

Ahora a cada conjunto obtenido en el paso anterior, se calcula la desviación estándar.

Se selecciona el valor máximo de cada parámetro, ya que se desea maximizar la cantidad de instancias que alcancen el valor óptimo.

Ahora que se tienen los valores máximos de cada parámetro, se elige cuál parámetro tiene un mayor valor, ya que este parámetro será la base para encontrar un mejor desempeño.

**4.- Update.** Cuando se haya encontrado al parámetro que aporte un mayor rendimiento al desempeño del algoritmo, se procederá a reducir el rango de este parámetro con la finalidad de reducir el espacio de búsqueda sobre ese parámetro y explorar en los restantes.

El enfoque de configuración iterará hasta haber disminuido el rango de cada parámetro al tamaño  $R_i^{\min}$  o hasta encontrar una configuración que obtenga el desempeño deseado.

# Capítulo 5

## V. EXPERIMENTACIÓN Y RESULTADOS

---

### 5.1 EVALUACIÓN INICIAL DE M-GGA-CGT

Para la experimentación se hizo uso de un conjunto de cinco instancias estándar tomadas de manera aleatoria de cinco grupos diferentes de TSP (Reinelt, 95), las cuales fueron resueltas con el clásico algoritmo NSGA-II (Deb, 2002) y con el algoritmo propuesto M-GGA-CGT. Los parámetros considerados en la experimentación fueron los siguientes: porcentaje de cruza = 0.2, porcentaje de mutación = 0.83, tamaño de la población = 100, número de generaciones = 30. Además, se considera que, a excepción del vértice origen y destino, un vértice solo debe estar dentro de una ruta.

La Figura 5 muestra una instancia en la que, en ambos objetivos, M-GGA-CGT obtuvo mejores resultados. Dado que la función objetivo  $f_1(S)$  es de minimización y la función objetivo  $f_2(S)$  es de maximización, el frente de pareto óptimo debe estar orientado hacia la región superior izquierda, tal como lo hace M-GGA-CGT.

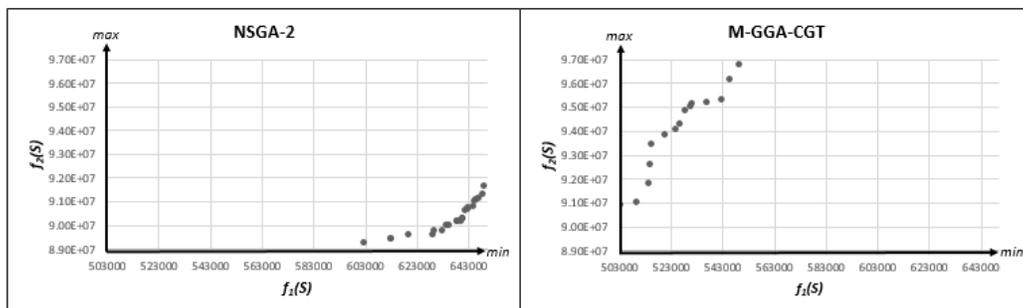


Figura 5. Comparación del mejor frente encontrado para la instancia ali532.

En la Tabla 1 se muestran los resultados de la experimentación. Se utilizó como indicadores de calidad la dominancia de Pareto (Deb, 2002) y la Distancia Generacional (Robič, 2005)(DG). Para obtener los resultados de dominancia de Pareto se compararon las soluciones obtenidas por cada algoritmo y se contabilizó cuántas se mantuvieron como no-dominadas. Por otro lado, la DG refleja la distancia existente entre un conjunto de soluciones no-dominadas y un conjunto representativo del mejor frente de Pareto aproximado (FP aproximado). En este caso, el FP aproximado está formado por la unión de las soluciones no-dominadas de ambos algoritmos.

**TABLA 4. COMPARACIÓN DE LOS RESULTADOS OBTENIDOS ENTRE NSGA-II Y M-GGA-CGT**

No.	Instancia	Algoritmo	Tamaño del conjunto de soluciones	Soluciones no dominadas en $(A \cup B)$	Distancia generacional	Distancia euclidiana (promedio)
1	rat575	NSGA-II	11	11	0	1.86
		M-GGACGT	18	18	0	1.02
2	u574	NSGA-II	13	13	0	1.87
		M-GGACGT	22	22	0	1.06
3	pcb442	NSGA-II	11	7	0.05	1.69
		M-GGACGT	17	17	0	0.59
4	ali532	NSGA-II	23	0	-	-
		M-GGACGT	15	15	0	0.82
5	d493	NSGA-II	10	6	1.2	1.41
		M-GGACGT	13	1	2.95	1.7

Nota A es el conjunto de soluciones obtenidas por NSGA-II; B es el conjunto obtenido por M-GGA-CGT

Se aprecia que, en las primeras dos instancias ambos algoritmos tuvieron buenos resultados en distancia generacional (Columna 5), pero M-GGA-CGT fue mejor en dominancia (Columna 4). En las instancias 3 y 4, M-GGA-CGT obtuvo mejores resultados que NSGA-II tanto en dominancia como en DG. Cabe mencionar que en

la instancia 4 ninguna solución de NSGA-II permaneció como no-dominada, por lo tanto no fue posible calcular su DG. Fue solo en la instancia 5 donde el NSGA-II obtuvo mejores resultados.

## 5.2 CARACTERIZACIÓN DEL DESEMPEÑO

Para la experimentación se hizo uso de un conjunto de 42 instancias estándar, estas fueron resueltas con el clásico algoritmo NSGA-II (Deb, 2002) y con el algoritmo propuesto M-GGA-CGT.

Los parámetros considerados en la experimentación para el algoritmo M-GGA.CGT fueron los siguientes: porcentaje de cruza = 0.2, porcentaje de mutación = 0.83, tamaño de la población = 100, número de generaciones = 30. En cuanto al algoritmo NSGA-II [4], los parámetros fueron: porcentaje de cruza = 0.8, porcentaje de muta = 0.1, tamaño de la población = 100, número de generaciones = 30.

Grupo de instancias	Soluciones óptimas	
	NSGA-II	M-GGA-CGT
ch	8.5	14
gr	10	14.3
kroA	6.6	12
kroB	8.3	9
lin	5.5	6.5
pr	6.8	11.2
rat	8.5	11

**TABLA 5. COMPARACIÓN DE LA CANTIDAD DE SOLUCIONES PROMEDIO GENERADAS POR EL ALGORITMO M-GGA-CGT Y EL ALGORITMO NSGA-II**

Como se aprecia en la Tabla 5, el algoritmo M-GGA-CGT siempre generó un conjunto mayor de soluciones.

### 5.2.1 Caracterización del Problema

Se aplicaron métricas que nos permite conocer el comportamiento del algoritmo, en base a los resultados se logró obtener el siguiente análisis de co-relación (Tabla 6).

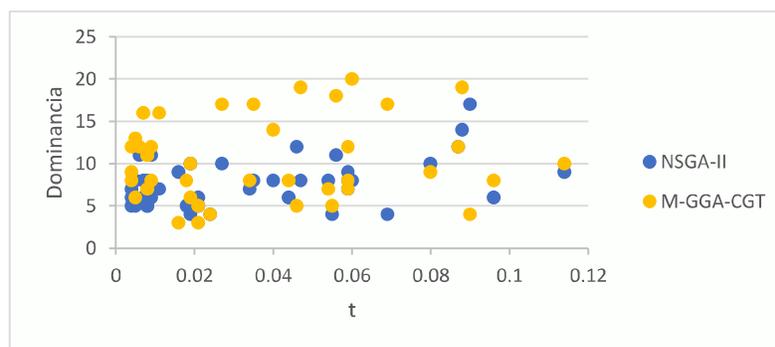
El análisis muestra que existe una relación entre las características de las instancias y el desempeño del algoritmo NSGA-II.

	<i>Uniformidad</i>	<i>t</i>	<i>d</i>	<i>Rango</i>	<i>Promedio</i>
Soluciones Generadas NSGA2	0.16120342	0.42286971	0.52156191	-0.11900813	-0.11403058
Soluciones Generadas GGACGT	-0.16287694	0.08303946	0.04245882	0.01381723	0.01265609
Soluciones Generadas Unión	-0.05470307	0.27150702	0.2861095	-0.04616756	-0.04471091
Dominancia NSGA2	0.16120342	0.42286971	0.52156191	-0.11900813	-0.11403058
Dominancia GGACGT	-0.16287694	0.08303946	0.04245882	0.01381723	0.01265609
Distancia Euclidiana NSGA2	-0.21930925	0.01773821	0.01265989	0.36113163	0.4231053
Distancia Euclidiana MGGACGT	-0.05838309	-0.01933068	0.01816128	0.1109613	0.1452751

**TABLA 6. ANÁLISIS DE CORRELACIÓN REALIZADO**

Para el algoritmo M-GGA-CGT las características de las instancias no parecen afectar el desempeño. Esto puede considerarse algo positivo, pues el algoritmo independientemente de la estructura de las instancias logra explorar y explotar de manera eficiente el espacio de búsqueda. Lo anterior confirma la generalidad del algoritmo M-GGA-CGT para solucionar de manera eficiente problemas de agrupación.

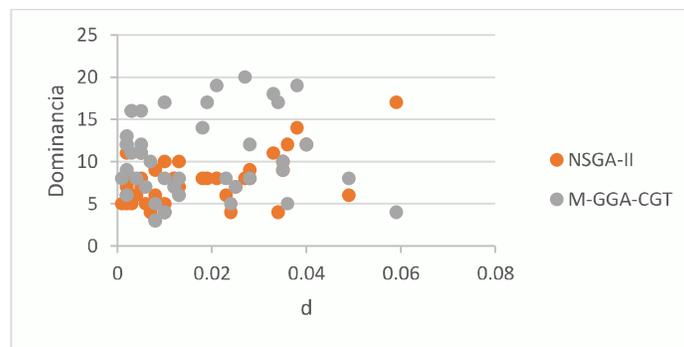
Los resultados obtenidos muestran que M-GGA-CGT es una estrategia más robusta y que permite realizar una mejor exploración del espacio de búsqueda que el algoritmo NSGA-II.



*Gráfica 1 Promedio de las distancias estandarizadas y dominancia*

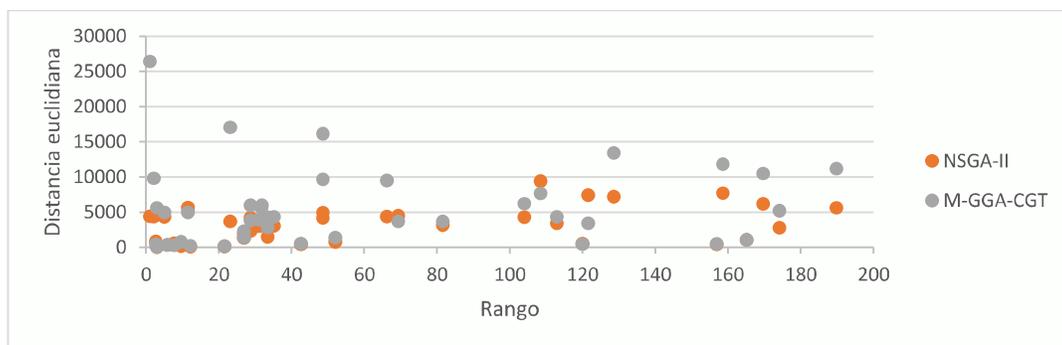
La Gráfica 1 Muestra que la dominancia del NSGA-II parece aumentar con el promedio de las distancias estandarizadas, en cambio M-GGA-CGT muestra un comportamiento similar en todas las instancias.

La dominancia del NSGA-II parece aumentar con la desviación de las distancias estandarizadas. Por otro lado, M-GGA-CGT muestra un comportamiento similar en todas las instancias. Lo anterior se aprecia en la Gráfica 2.



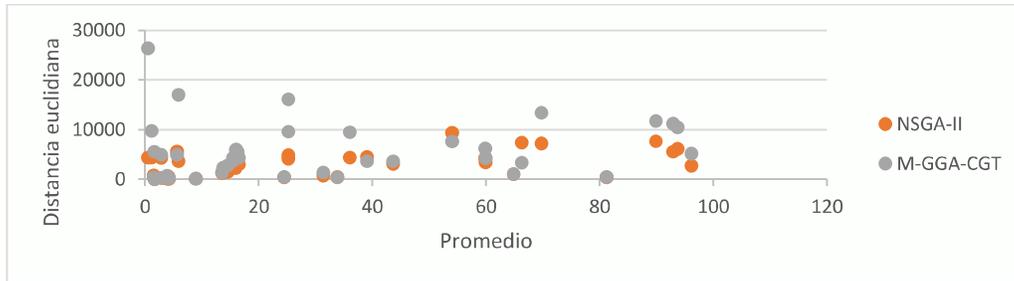
Gráfica 2 Desviación de las distancias estandarizadas y DOMINANCIA

En la Gráfica 3, M-GGA-CGT muestra un comportamiento similar en todas las instancias. La distancia euclidiana del NSGA-II parece aumentar con el rango de las distancias.



Gráfica 3 Distancia euclidiana y rango

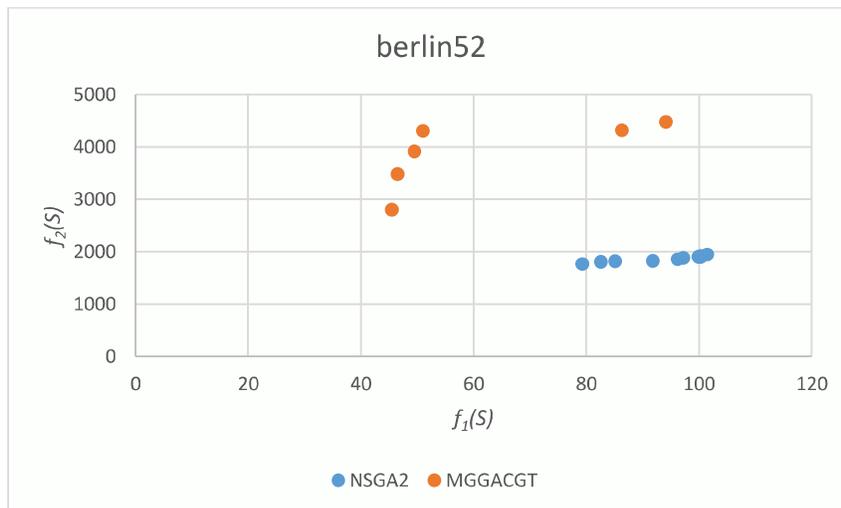
M-GGA-CGT muestra un comportamiento similar en todas las instancias, como se muestra en la Gráfica 4. En cambio, en NSGA-II la distancia euclidiana parece aumentar con el promedio de las distancias.



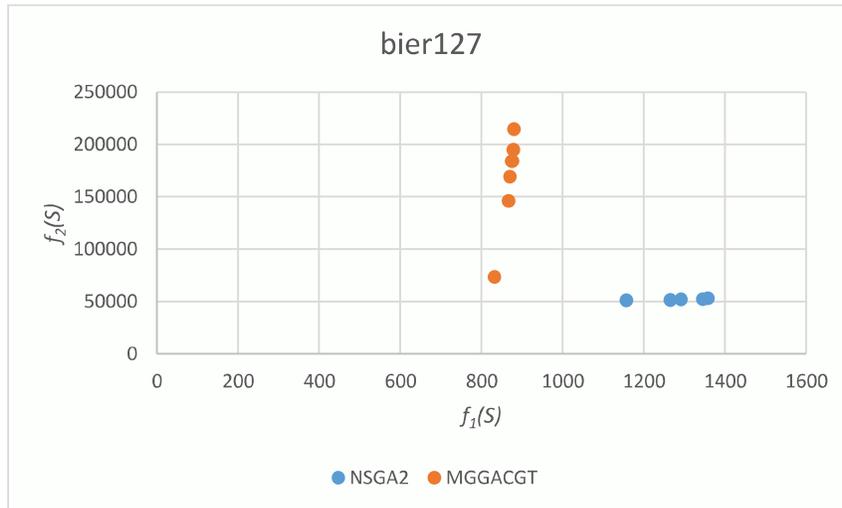
Gráfica 4 Distancia euclidiana Y promedio

### 5.2.2 Caracterización de la salida

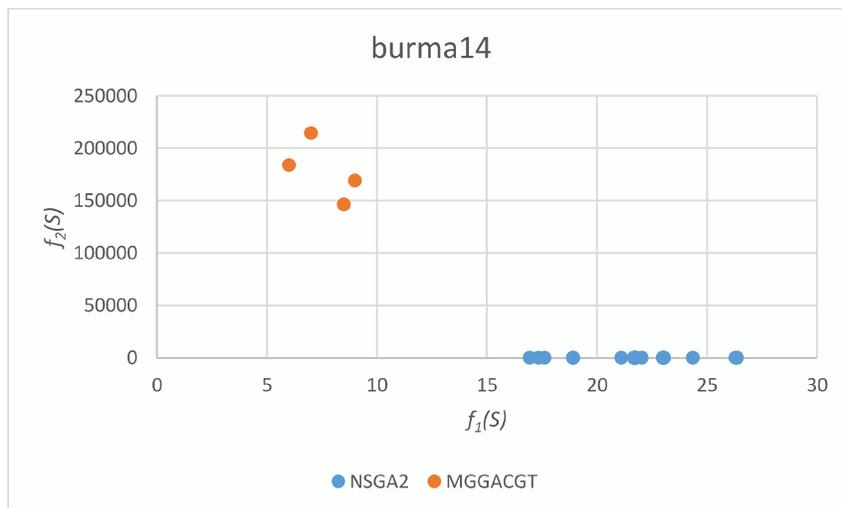
Se realizó la experimentación con un grupo de 42 instancias obteniendo resultados satisfactorios al comparar la eficiencia contra el algoritmo NSGA2. A continuación se muestran gráficas para demostrar la efectividad del algoritmo M-GGA-CGT.



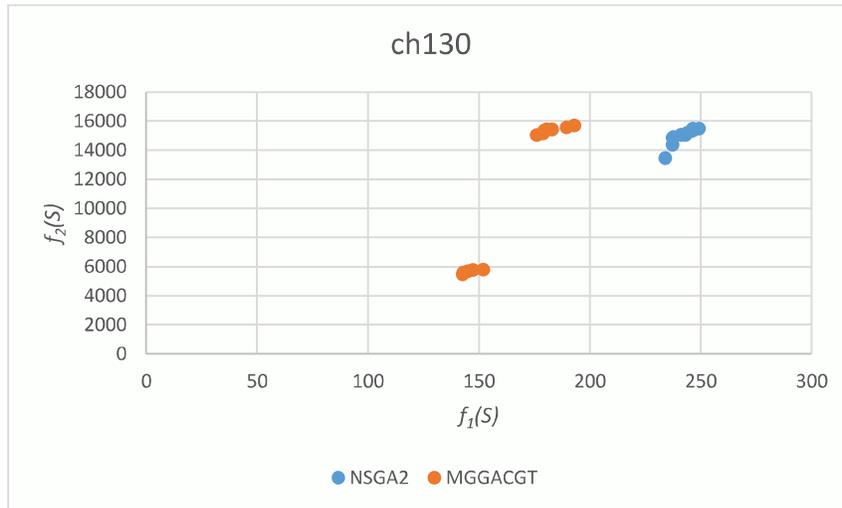
Gráfica 5 Comparación de resultados obtenidos con la instancia berlin52.



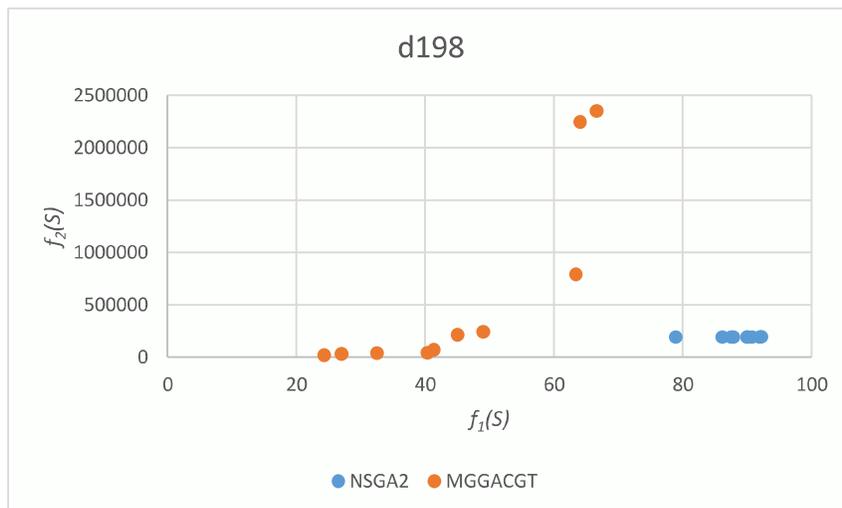
Gráfica 6 Comparación de resultados obtenidos con la instancia bier127



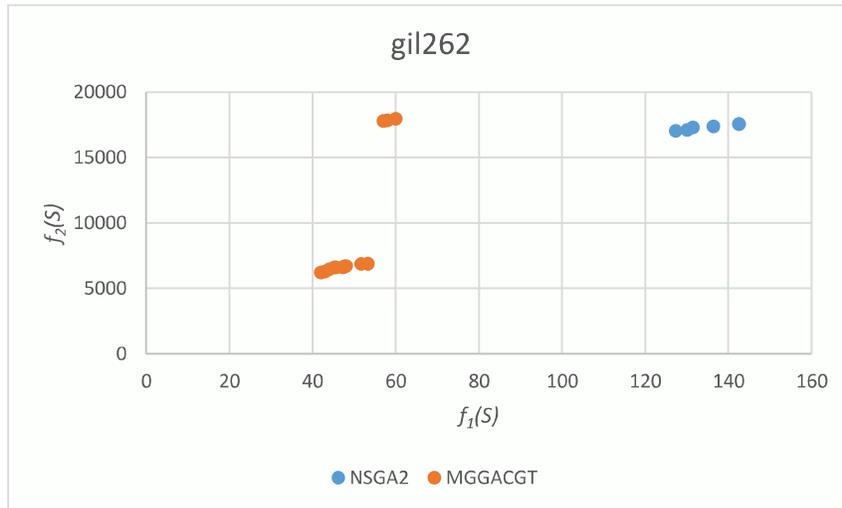
Gráfica 7 Comparación de resultados obtenidos con la instancia burma14



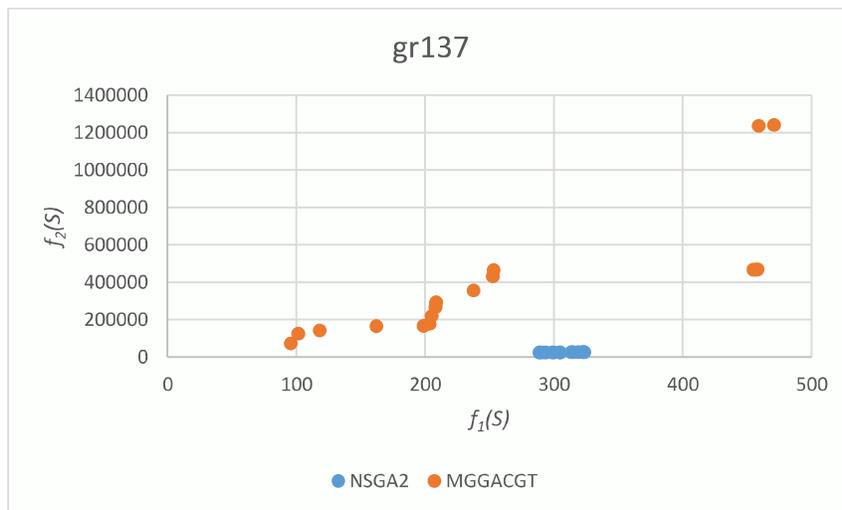
Gráfica 8 Comparación de resultados obtenidos con la instancia ch130.



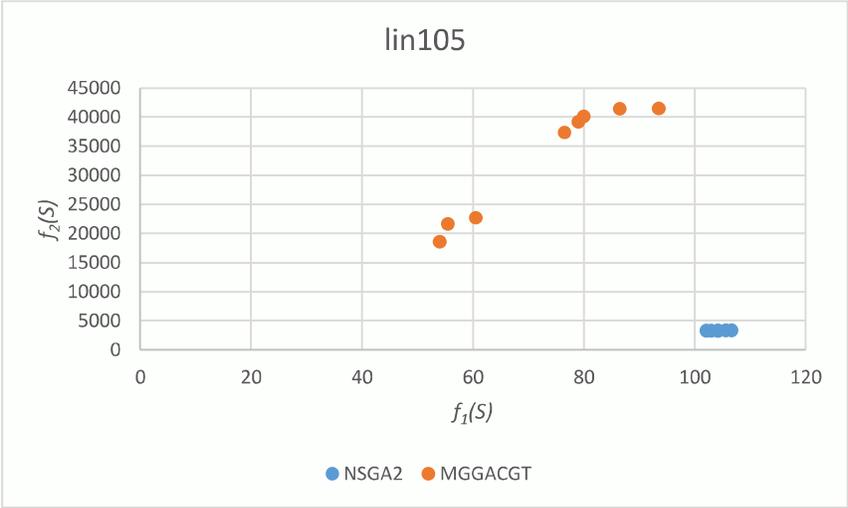
Gráfica 9 Comparación de resultados obtenidos con la instancia d198.



Gráfica 10 Comparación de resultados obtenidos con la instancia gil162.



Gráfica 11 Comparación de resultados obtenidos con la instancia gr137.



Gráfica 12 Comparación de resultados obtenidos con la instancia lin105.

# Capítulo 6

## VI. CONCLUSIONES

---

En este capítulo se presentan las conclusiones y los resultados finales obtenidos como producto de la investigación desarrollada y se dan sugerencias para el desarrollo de trabajos futuros.

### 6.1 CONCLUSIONES

En el trabajo de investigación presentado en esta tesis se abordó la adaptación de un algoritmo del estado del arte, además se estudió el proceso de solución de problemas de optimización NP-duros seleccionando el más acercado al problema que resuelve el algoritmo original.

Una vez estudiado el algoritmo original se procedió a desarrollar un nuevo algoritmo, el cual tuviera todas las características buscando siempre buenos resultados, para ello se compararon los resultados obtenidos contra otro algoritmo fuerte del estado del arte, el NSGA2 (Deb, 2002).

El algoritmo propuesto fue evaluado al aplicarse al problema Path Dissimilarity, el cual es un problema de rutas. Los resultados experimentales obtenidos mostraron que el algoritmo rediseñado supera las soluciones del algoritmo del estado del arte, NSGA2.

Entre las principales aportaciones de este trabajo se encuentran:

1. Se desarrolló la adaptación de un algoritmo del estado del arte, el GGA-CGT (Quiroz, 2014).

2. Se realizó un estudio sobre la estructura del problema Path Dissimilarity y relación con el problema de empaqueo de objetos en contenedores, el cual es resuelto por el algoritmo original.
3. Se analizó y seleccionó un conjunto de índices de caracterización que pudieran ser aplicados al problema Path Dissimilarity.
4. Se implementó una metodología que permite configurar los parámetros de entrada del algoritmo.
5. Una vez terminado el algoritmo se aplicó a 42 instancias, obteniendo resultados satisfactorios al compararlos con el NSGA2. Demostrando que el algoritmo generado es competitivo.

## **6.2 TRABAJO FUTURO**

Para dar continuidad al trabajo de investigación presentado se proponen los siguientes trabajos:

- a. Aplicar el algoritmo generado en esta tesis a otro gran número de instancias.
- b. Se considera posible aplicar el algoritmo M-GGA-CGT a otros problemas de agrupación, solo realizando unas pequeñas adaptaciones.

## VII. REFERENCIAS BIBLIOGRÁFICAS

---

- (Álvarez, 2006) Álvarez V. (2006). *Modelo para representar la Complejidad del problema y el desempeño de algoritmos*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México.
- (Baesler, 2008) Baesler, F. (2008). Introducción de elementos de memoria en el método SIMULATED ANNEALING para resolver problemas de programación multiobjetivo de máquinas paralelas. *INGENIARE Revista Chilena de Ingeniería*, 428-437.
- (Baños, 2006) Baños N.R. (2006). *Metaheurísticas Híbridas para Optimización Mono-objetivo y Multiobjetivo. Paralelización y Aplicaciones*. Tesis de doctorado. Universidad de Almería, España.
- (Bartz, 2014) Bartz-Beielstein, T., & Preuß, M. (2014). Experimental analysis of optimization algorithms: Tuning and beyond. In *Theory and Principled Methods for the Design of Metaheuristics* (pp. 205-245). Springer Berlin Heidelberg.
- (Bechikh, 2013) Bechikh, Slim. (2013). *Incorporating Decision Maker's Preference Information in Evolutionary Multi-objective Optimization*. Tesis de doctorado, High Institute of Management of Tunis, University of Tunis, Tunisia.
- (Bielza, 2010) Bielza, C., del Pozo, J. F., Larrañaga, P., & Bengoetxea, E. (2010). Multidimensional statistical analysis of the parameterization of a genetic algorithm for the optimal ordering of tables. *Expert Systems with Applications*, 37(1), 804-815.
- (Camacho, 2015) Camacho, J., Pérez-Villegas, A., Rodríguez-Gómez, R. A., & Jiménez-Mañas, E. (2015). Multivariate Exploratory Data Analysis (MEDA) Toolbox for Matlab. *Chemometrics and Intelligent Laboratory Systems*, 143, 49-57.
- (Coello, 2007) Coello, G. B. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer.
- (Cruz, 2004) Cruz L. (2004). *Caracterización de Algoritmos Heurísticos Aplicados al Diseño de Bases de Datos Distribuidas*. Tesis de doctorado, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Morelos, México.
- (Deb, 2002) Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2), 182-197.
- (Duarte, 2007) Duarte, A. P. (2007). *METAHEURÍSTICAS*. S.L. - DYKINSON.
- (Ferat, 2007) Ferat E. (2007). *Formación Dinámica de Regiones de Dominación de Algoritmos Heurísticos*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México.
- (García, 2004) García N. (2004). *Análisis de las Propiedades del Problema de la Distribución de Bases de Datos y su Influencia en el Desempeño de Algoritmos Heurísticos*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México.
- (García, 2010) García R. (2010). *Hiper-heurístico para resolver el problema de cartera de proyectos sociales*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México.
- (Jiang, 2014) Jiang, S., Ong, Y. S., Zhang, J., & Feng, L. (2014). Consistencies and contradictions of performance metrics in multiobjective optimization. *IEEE Transactions on Cybernetics*, 44(12), 2391-2404.
- (Krause, 2011) Krause, L. H., Franz, A., & Stevenson, A. (2011). On the application of Exploratory Data Analysis for characterization of space weather data sets. *Advances in Space Research*, 47(12), 2199-2209.
- (Landeró, 2004) Landeró V. (2004). *Análisis de las Regiones de Dominación de Algoritmos Heurísticos Aplicados al Diseño de Bases de Datos Distribuidas*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México.
- (Landeró, 2008) Landeró V. (2008) *Desarrollo de un Método Formal que Muestre la Interrelación entre las Características de un Conjunto de Casos y las de un Algoritmo que los Resuelve Eficientemente, para el Problema de Distribución de Objetos en Contenedores*. Tesis

- de doctorado, Dpto. de Ciencias Computacionales, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Mor.
- (Martí, 2009) Martí, R., Velarde, J. L. G., & Duarte, A. (2009). *Heuristics for the bi-objective path dissimilarity problem*. *Computers & Operations Research*, 36(11), 2905-2912.
- (Martí, 2012) Martí R, Campos, V, Resende, MGC, Duarte A. (2012). *Multiobjective GRASP with Path Relinking*. Technical Report.
- (Martí, 2015) Martí, R., Campos, V., Resende, M. G., & Duarte, A. (2015). Multiobjective GRASP with path relinking. *European Journal of Operational Research*, 240(1), 54-71.
- (Mendes, 2016) Mendes, R. S., Miranda, D. S., Wanner, E. F., Sarubbi, J. F., & Martins, F. V. (2016). *Multiobjective Approach to the Vehicle Routing Problem with Demand Responsive Transport*.
- (Pérez, 2007) Pérez V. (2007). *Modelado Causal del Desempeño de Algoritmos Metaheurísticos en Problemas de Distribución de Objetos*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México.
- (Quiroz, 2004) Quiroz M. (2004). *Caracterización de factores de desempeño de algoritmos de solución de BPP*. Tesis de doctorado. Instituto Tecnológico de Ciudad Madero, Tamaulipas, México.
- (Quiroz, 2014) Quiroz M. (2014). *Caracterización del proceso de optimización de algoritmos heurísticos aplicados al problema de empaqueo de objetos en contenedores*. Tesis de doctorado. Instituto Tecnológico de Tijuana, Baja California Norte, México.
- (Reinelt, 95) Reinelt, Gerhard. TSLIB. <http://comopt.ifi.unireidelberg.de/software/TSPLIB95/>
- (Rubič, 2005) Rubič, T., & Filipič, B. (2005). DEMO: Differential evolution for multiobjective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 520-533). Springer Berlin Heidelberg.
- (Santiago, 2004) Santiago E.(2004). *Modelo Estadístico para la Caracterización de Casos del Problema del Diseño de la Distribución*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México.
- (TALBI, 2009) TALBI, El-Ghazali. (2009). *Metaheuristics: from design to implementation*. John Wiley & Sons.
- (Tukey, 1977) Tukey, J. W. (1977). *Exploratory Data Analysis*. Pearson.
- (Van, 1998) Van Veldhuizen, D. A., & Lamont, G. B. (1998). *Multiobjective evolutionary algorithm research: A history and analysis*. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.
- (Veček, 2014) Veček, N., Mernik, M., & Črepinšek, M. (2014). A chess rating system for evolutionary algorithms: A new method for the comparison and ranking of evolutionary algorithms. *Information Sciences*, 277, 656-679.
- (Xie, 2013) Xie, D., Ding, L., Hu, Y., XIE, C., WANG, S., & Jiang, L. (2013). *A Multi-Algorithm Balancing Convergence and Diversity for Multi-Objective Optimization*. *Journal of Information Science & Engineering*, 29(5).
- (Yen, 2014) Yen, G.G.; Zhenan He. (2014). Performance Metric Ensemble for Multiobjective Evolutionary Algorithms," *Evolutionary Computation*, IEEE Transactions on, vol.18, no.1, pp.131,144.
- (Zitzler, 1999) Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *Evolutionary computation, IEEE transactions on*, 3(4), 257-271.
- (Zitzler, 2000) Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2), 173-195.