

**MÉTODOS COMPUTACIONALES PARA LA SOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN
BAJO INCERTIDUMBRE Y PROBLEMAS INVERSOS, DESDE LA PERSPECTIVA
INTERVALO-VALUADA**

Andrés Felipe Cano Cadavid
Código 200919000016

ESCUELA DE CIENCIAS Y HUMANIDADES
DEPARTAMENTO DE CIENCIAS BÁSICAS
MAESTRÍA EN MATEMÁTICAS APLICADAS
MEDELLÍN

2011

**MÉTODOS COMPUTACIONALES PARA LA SOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN
BAJO INCERTIDUMBRE Y PROBLEMAS INVERSOS, DESDE LA PERSPECTIVA
INTERVALO-VALUADA**

**Trabajo de investigación presentado como requisito para optar al título de Magíster en
Matemáticas Aplicadas**

Andrés Felipe Cano Cadavid

Director

**María Eugenia Puerta Yepes
Doctor en Ciencias Matemáticas**

ESCUELA DE CIENCIAS Y HUMANIDADES

DEPARTAMENTO DE CIENCIAS BÁSICAS

MAESTRÍA EN MATEMÁTICAS APLICADAS

MEDELLÍN

2011

Nota de aceptación

Coordinador de la Maestría

Director del proyecto

Medellín, 11 de marzo del 2011:

Agradecimientos

Quiero expresar los más sinceros agradecimientos, a mis asesora María Eugenia Puerta Yepes por su tiempo, su dedicación, orientación y enseñanzas; a mi familia por estar siempre de forma incondicional a mi lado; a la Universidad EAFIT por apoyarme; a mis amigos y todas aquellas personas que de alguna u otra forma participaron en la realización de este trabajo.

Índice general

Resumen	1
Introducción	3
1. Preliminares	5
1.1. Notación	5
1.2. Optimización Multi-objetivo Intervalo-valuada	5
1.2.1. Conceptos Básicos	5
1.2.2. Funciones Multi-Intervalo-Valuadas	7
1.2.3. Diferenciabilidad en Funciones Multi-intervalo-valuadas	8
1.2.4. Problema de Optimización y Concepto de Solución	9
1.2.5. Condiciones Karush-Kuhn-Tucker (KKT) de Optimalidad	11
1.2.5.1. Condiciones KKT para el caso continuo débilmente diferenciable .	11
1.2.5.2. Condiciones KKT para el caso continuo H-diferenciable	16
1.2.5.3. Ejemplos Numéricos	17
1.3. Métodos computacionales	19
1.3.1. Algoritmos Evolutivos	20
1.3.1.1. Conceptos Básicos de Algoritmos Evolutivos	20
1.3.1.2. Clasificación de Técnicas de Búsqueda y Decisión	23
1.3.1.2.1. Técnicas <i>A Priori</i>	23
1.3.1.2.2. Técnicas <i>Progresivas</i>	23
1.3.1.2.3. Técnicas <i>A Posteriori</i>	23
1.3.1.3. Técnicas para Preservación de la Diversidad	24
1.3.1.4. Elitismo	25
1.3.1.5. Representación del Cromosoma	25

1.3.1.5.1.	Operadores de Cruce	25
1.3.1.5.2.	Operadores de Mutación	26
1.3.1.6.	Técnicas para el Manejo de Restricciones	27
1.3.1.6.1.	Funciones de Penalización:	27
1.3.1.6.2.	Separación de las Restricciones y Objetivos	29
1.3.2.	Strength Pareto Evolutionary Algorithm - SPEA2	29
1.3.2.1.	Función de Ajuste y Preserveración de la Diversidad	30
1.3.2.2.	Selección Siguiete Generación	31
1.3.3.	MultiObjective Particle Swarm Optimization - MOPSO	32
2.	Algoritmos Propuestos	35
2.1.	Strength Pareto Evolutionary Algorithm for Interval-valued Problems - SPEA2I	35
2.1.1.	Representación del Cromosoma	36
2.1.2.	Función de Ajuste y Preserveración de la Diversidad	36
2.1.3.	Selección Siguiete Generación	36
2.1.4.	Algoritmo Implementado	36
2.2.	Multi-Objective Interval-Valued Particle Swarm Optimization - MOIPSO	37
2.3.	Ejemplos Numéricos	38
3.	Problema Inverso	43
3.1.	Conceptos Básicos	43
3.2.	Criterios de Identificación	44
3.2.1.	Minimizar Distancias	45
3.2.2.	Markov	45
3.2.3.	Maximizar Verosimilitud	46
3.3.	Métodos de Solución	46
3.3.1.	Métodos Numéricos	47
3.3.1.1.	Gauss - Newton (GNA)	47
3.3.1.2.	Levenberg - Marquardt (LMA)	47
3.3.2.	Métodos Heurísticos	47
3.4.	Ejemplos Numéricos	48
3.4.1.	Estimación de las Coordenadas del Epicentro de un Evento Sísmico	48
3.4.2.	Medición de la Aceleración Gravitacional	49

4. Aplicaciones	53
4.1. Modelo Logístico	53
4.1.1. Aplicación del Modelo Logístico a los Censos de Medellín	55
4.1.2. Aplicación del Modelo Logístico a los Datos de Rangos de Edad para Antioquia	58
Conclusiones	61
Problemas abiertos	63

Índice de cuadros

2.1. Resultados obtenidos con los algoritmos propuestos para los ejemplos del capítulo 1 .	38
3.1. Diez mejores resultados de las Coordenadas del Epicentro	49
3.2. Diez mejores resultados para la Velocidad Inicial y la Gravedad	50
4.1. Datos de población para la ciudad de Medellín. t_i : Año censal contado a partir del año base (1912)	56
4.2. Datos de comparación entre los datos del censo, los datos estimados por el modelo y los datos estimados en [18]	57
4.3. Datos de comparación entre los datos del censo, los datos estimados por el modelo y los datos estimados en [18]	59
4.4. Datos de población por quinquenal para el Departamento de Antioquia	59
4.5. Parámetros de ejecución para el SPEA2I en el problema de estimación de parámetros del modelo logístico a los datos de quinquenales para Antioquia	59
4.6. Datos de población por quinquenal para el Departamento de Antioquia obtenidos con el modelo	60

Índice de figuras

1.1. Modelo estructural de una cercha de diez barras	19
1.2. Ilustración del método de truncamiento utilizado en SPEA2. A la derecha, un conjunto no-dominado. A la izquierda, se representan las soluciones que se eliminan por el operador de truncamiento (suponiendo que $N = 5$).	32
2.1. Función de Rastrigin	39
2.2. Función de Rosenbrock	40
2.3. Función de Schwefel	40
2.4. Función de Michalewicz	41
3.1. Valor Promedio de las Coordenadas de Epicentro	49
3.2. Valores Promedio de la Velocidad Inicial y Gravedad	51
4.1. Modelo Logístico: Comparación de los Datos Observados y los Datos Predichos . . .	57
4.2. Modelo Logístico: Comparación de los Datos del Censo, los Datos Predichos por el Modelo y los Datos Predichos en [18]	58

Resumen

En este trabajo se presenta la manera como algunos de los problemas de optimización multi-objetivo bajo incertidumbre se pueden formular a través de la teoría de optimización intervalo-valuada y se proponen dos algoritmos para resolverlos. Además, se realiza una aplicación para un problema real, el modelo logístico poblacional para la ciudad de Medellín y Antioquia, donde se estiman los parámetros a través de los algoritmos propuestos con base en los datos del DANE.

OPTIMIZACIÓN INTEVALO-VALUADA, CONVEXIDAD Y DIFERENCIABILIDAD, OPTIMIZACIÓN MULTI-OBJETIVO, MÉTODOS COMPUTACIONALES

Introducción

La incertidumbre en los problemas de optimización está ligada a la información, ya sea de los parámetros, las variables o las funciones. Esta incertidumbre puede provenir por ausencia, ambigüedad, inconsistencia, error, imprecisión, entre otros, de la información. Para resolver este tipo de problemas se han planteado diferentes perspectivas, las principales son la Optimización Estocástica y la Optimización Fuzzy o Difusa; la primera maneja la incertidumbre a través de la teoría de probabilidades, donde se requiere conocer la distribución de los datos y la segunda, a través de la teoría fuzzy o difusa, por lo tanto se requiere conocer la función de pertenencia de los datos.

La teoría intervalo-valuada es una perspectiva diferente para manejar la incertidumbre en estos problemas, en ella sólo es necesario conocer el intervalo donde se encuentran los datos. Esta teoría ha sido fuertemente desarrollada desde los años noventa con los trabajos de Ishibuchi y Tanaka [10] y posteriormente por Wu [30], [31] y [32], aunque sus orígenes se remontan a los trabajos de Moore [16] en los años setenta. Esta teoría ha sido aplicada en la solución de diferentes tipos de problemas, principalmente en optimización estructural, debido a la naturaleza misma de los parámetros y variables que tienen este tipo de problemas.

Por otra parte, la teoría de optimización intervalo-valuada ha logrado desarrollos teóricos tanto para problemas con funciones mono-objetivo como problemas con funciones multi-objetivo, los cuales requieren que dichas funciones tengan algunas características especiales, tales como convexidad y diferenciabilidad, pero, dado que en los problemas prácticos no siempre se satisfacen dichas condiciones, se hace necesario recurrir a métodos numéricos o computacionales de búsqueda de la solución, o las soluciones, más laxos en sus condiciones de aplicación que un método analítico. Además, se debe resaltar que algunos de los métodos computacionales han demostrado tener gran capacidad de exploración del espacio de búsqueda, pueden ser robustos y no tener problemas de sensibilidad.

Aunque existen muchos desarrollos de métodos numéricos y computacionales para resolver problemas de optimización multi-objetivo, son escasos aquellos que están diseñados para resolver problemas de optimización multi-objetivo que involucran restricciones como incertidumbre al mismo tiempo; por este motivo en este trabajo se proponen dos métodos que involucran estas tres características desde la perspectiva intervalo-valuada.

En este trabajo se presenta la manera como los problemas de optimización bajo incertidumbre se pueden formular a través de la teoría de optimización intervalo-valuada y se proponen dos algoritmos para resolverlos. En el primer capítulo se presentan los conceptos necesarios para entender el problema de optimización multi-objetivo intervalo-valuado y las técnicas de solución analíticas; allí se muestran las definiciones y teoremas necesarios de la teoría intervalo-valuada para solución de problemas de optimización; además, se presentan los conceptos básicos de los algoritmos evolutivos y Particle Swarm Optimization - PSO -, donde para cada uno se explica el algoritmo base. En el siguiente capítulo se presentan los algoritmos propuestos, donde se incorpora la incertidumbre y las restricciones a los algoritmos presentados en el primer capítulo. Estos algoritmos pueden ser adaptados para la solución de un tipo especial de problemas de optimización como los problemas inversos, los cuales son muy utilizados en Geotecnia y Geofísica para identificación de parámetros, por tal motivo el capítulo tres se enfoca en los conceptos básicos de este tipo de problemas. Por último, se presenta un capítulo donde se realiza una aplicación para un problema real, el modelo logístico poblacional para la ciudad de Medellín y Antioquia, donde se estiman los parámetros a través de los algoritmos propuestos con base en los datos del DANE.

Capítulo 1

Preliminares

En este capítulo exponen los elementos que sirven de base para la propuesta central de este trabajo. En esta dirección, lo primero será describir los fundamentos y resultados matemáticos de la optimización multi-objetivo con incertidumbre modelada a través de intervalos, en donde los resultados a tener en cuenta están ligados a los artículos [30] y [31]. Una segunda componente de este capítulo la conforma el estado del arte en heurísticos para optimización multi-objetivo sin consideraciones de incertidumbre, pero que serán la fuente para proponer dos algoritmos que sí consideran incertidumbre del tipo mencionado antes en los problemas de optimización multi-objetivo.

1.1. Notación

La notación utilizada en este trabajo será la habitual en Matemáticas. Se denota el cuerpo de números reales por \mathbb{R} , el espacio euclidiano de dimensión n de los números reales por \mathbb{R}^n , los conjuntos o espacios vectoriales con letras mayúsculas, los elementos de los conjuntos con letras minúsculas, las constantes con letras griegas como λ, μ , entre otros. Se utilizan las letras minúsculas f, g y h para denotar funciones; las letras mayúsculas A, B y C para hacer referencia a elementos del conjunto intervalo-valuado. En general, la notación será clara a partir de la sección de conceptos básicos.

1.2. Optimización Multi-objetivo Intervalo-valuada

En esta sección se presenta, para claridad del contenido del trabajo, los conceptos de orden, distancia, métrica y óptimo de Pareto, entre otros, considerados en la teoría multi-intervalo-valuada. Estos conceptos fueron aplicados en la implementación de los algoritmos propuestos para poder determinar cuál es el conjunto solución del problema.

1.2.1. Conceptos Básicos

Definición 1.2.1. *Al conjunto de todos los arreglos de m intervalos cerrados, acotados y convexos en \mathbb{R} los denotamos por $I^m(\mathbb{R})$. Si $A \in I^m(\mathbb{R})$, se puede describir a A así:*

$$A = \begin{bmatrix} [a_1^L, a_1^U] \\ \dots \\ [a_m^L, a_m^U] \end{bmatrix}$$

donde a_i^L y a_i^U son los límites inferior y superior del intervalo i , respectivamente, es decir $a_i^L \leq a_i^U$. En el conjunto $I^m(\mathbb{R})$ se puede definir una aritmética que básicamente proviene de la aritmética de intervalos en el contexto de teoría de errores, desarrollada por Moore en [16]. A continuación se presentan las operaciones básicas:

Sean A y $B \in I(\mathbb{R})$ se definen las operaciones aritméticas básicas así:

- **Suma:** $A + B = [a^L, a^U] + [b^L, b^U] = [a^L + b^L, a^U + b^U]$
- **Producto:** $A \times B = [a^L, a^U] \times [b^L, b^U] = [\min(a^L b^L, a^L b^U, a^U b^L, a^U b^U), \max(a^L b^L, a^L b^U, a^U b^L, a^U b^U)]$
- **División:** $A/B = [a^L, a^U] / [b^L, b^U] = [\min(a^L/b^L, a^L/b^U, a^U/b^L, a^U/b^U), \max(a^L/b^L, a^L/b^U, a^U/b^L, a^U/b^U)]$

Con base en estas operaciones para el conjunto $I(\mathbb{R})$ se definen la suma y producto por escalar en $I^m(\mathbb{R})$ de la siguiente manera: Sean A y $B \in I^m(\mathbb{R})$ y $\alpha \in \mathbb{R}$ definimos

- **Suma:** $A + B = \begin{bmatrix} A_1 + B_1 \\ \dots \\ A_m + B_m \end{bmatrix} = \begin{bmatrix} [a_i^L + b_i^L, a_i^U + b_i^U] \\ \dots \\ [a_m^L + b_m^L, a_m^U + b_m^U] \end{bmatrix}$
- **Producto por Escalar:** $\alpha A = \begin{bmatrix} \alpha A_1 \\ \dots \\ \alpha A_m \end{bmatrix} = \begin{cases} \begin{bmatrix} [\alpha a_1^L, \alpha a_1^U] \\ \dots \\ [\alpha a_m^L, \alpha a_m^U] \end{bmatrix} & \alpha \geq 0 \\ \begin{bmatrix} [\alpha a_1^U, \alpha a_1^L] \\ \dots \\ [\alpha a_m^U, \alpha a_m^L] \end{bmatrix} & \alpha < 0 \end{cases}$

Note que las operaciones definidas son clausurativas. Además de cumplir esta propiedad también cumplen las siguientes:

Proposición 1.2.1. : Sean A, B y $C \in I^m(\mathbb{R})$ y $\alpha, \beta \in \mathbb{R}^+$

La operación suma (+) cumple las siguientes propiedades:

- **Conmutativa:** $A + B = B + A$
- **Asociativa:** $A + (B + C) = (A + B) + C$
- **Elemento neutro:** $\mathbf{0} = \begin{bmatrix} [0, 0] \\ \dots \\ [0, 0] \end{bmatrix} \in I^m(\mathbb{R})$ tal que $A + \mathbf{0} = \mathbf{0} + A = A$

La operación producto por escalar (\bullet) cumple las siguientes propiedades:

- Asociativa: $\alpha(\beta A) = (\alpha\beta)A$
- Elemento neutro: $1 \in \mathbb{R}$ tal que $1A = A$
- Distributiva con respecto a la suma: $\alpha(A + B) = \alpha A + \alpha B$
- Distributiva con respecto a la suma de escalares: $(\alpha + \beta)A = \alpha A + \beta A$

Aunque el conjunto $I^m(\mathbb{R})$ tiene definidas las operaciones algebraicas adición (+) y producto por escalar (\bullet), las cuales cumplen la mayoría de las propiedades necesarias en un espacio vectorial, $I^m(\mathbb{R})$ no es un espacio vectorial porque con la operación suma no todos los elementos tienen inverso aditivo. Note que $\neg \forall A \in I^m(\mathbb{R}) \exists -A \in I^m(\mathbb{R}) : A + (-A) = 0$. Se resalta esta característica porque sólo sobre espacios vectoriales se puede definir el concepto de derivada de Frechet, la cual es necesaria para poder exponer los teoremas de existencia de solución en los problemas de optimización multi-intervalo valuada.

A continuación se presentará el concepto de diferencia de Hukuhara, el cual es necesario para después definir la diferenciabilidad en $I^m(\mathbb{R})$.

Definición 1.2.2. (Diferencia Hukuhara): Sean $Kc(\mathbb{R}) = \{C \in \mathbb{R} | C \text{ es conjunto compacto y convexo}\}$ y $A, B \in Kc(\mathbb{R})$, si existe $C \in Kc(\mathbb{R})$ tal que $A = B + C$, entonces C se llama la diferencia de Hukuhara, denotada por $C = A \ominus B$.

Al aplicar la definición anterior en el conjunto $I^m(\mathbb{R})$, sean $A, B \in I^m(\mathbb{R})$, la diferencia de Hukuhara entre A y B ($A \ominus B$) si existe, es el multi-intervalo $C = \begin{bmatrix} [c_1^L, c_1^U] \\ \dots \\ [c_m^L, c_m^U] \end{bmatrix}$ el cual está definido por

$$C = \begin{bmatrix} [A_1 \ominus B_1] \\ \dots \\ [A_m \ominus B_m] \end{bmatrix} = \begin{bmatrix} [a_1^L - b_1^L, a_1^U - b_1^U] \\ \dots \\ [a_m^L - b_m^L, a_m^U - b_m^U] \end{bmatrix}.$$

1.2.2. Funciones Multi-Intervalo-Valuadas

A continuación se definirá el concepto de función multi-intervalo-valuada y con base en ello y los conceptos vistos en la sección anterior se procederá a definir límite y continuidad, conceptos necesarios para poder definir la diferenciabilidad en este tipo de funciones.

Definición 1.2.3. (Función Multi-intervalo-valuada): Sea $f(x) : \mathbb{R}^n \rightarrow I(\mathbb{R})$ definida en el espacio euclideo \mathbb{R}^n con imagen en $I^m(\mathbb{R})$ es llamada función multi-intervalo-valuada, es decir, $f(x) = f(x_1, \dots, x_n)$ es un arreglo de intervalos cerrados en \mathbb{R} para todo $x \in \mathbb{R}^n$. También puede ser escrita como

$$f(x) = \begin{bmatrix} [f_1^L(x), f_1^U(x)] \\ \dots \\ [f_m^L(x), f_m^U(x)] \end{bmatrix}$$

donde $f_i^L(x)$ y $f_i^U(x)$ son funciones real-valuadas definidas en \mathbb{R}^n y satisfacen $f_i^L(x) \leq f_i^U(x)$ para todo $x \in \mathbb{R}^n$ y $i = 1, \dots, m$

Definición 1.2.4. (Límite): Sea $\left\{ A_n = \begin{bmatrix} [a_{n1}^L, a_{n1}^U] \\ \dots \\ [a_{nm}^L, a_{nm}^U] \end{bmatrix} \right\}$ y A un elemento en $I^m(\mathbb{R})$. Se dice que una secuencia de multi-intervalos $\{A_n\}$ converge a A , denotado por $\lim_{n \rightarrow \infty} A_n = A$, si y sólo si $\lim_{n \rightarrow \infty} a_{ni}^L = a_i^L$ y $\lim_{n \rightarrow \infty} a_{ni}^U = a_i^U$ para todo $i = 1, \dots, m$

Se definirá el concepto de métrica de Hausdorff, el cual es necesario para enunciar la definición de límite de una función multi-intervalo-valuada.

Definición 1.2.5. (Distancia Hausdorff): Sean X y Y dos subconjuntos de un espacio métrico (E, d) . Se define la métrica Hausdorff $d_H(X, Y)$ como:

$$d_H(X, Y) = \max \left\{ \sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \right\}$$

En particular para el conjunto $I(\mathbb{R})$ la métrica se define para dos elementos A y B que pertenezcan al conjunto así:

$$d_H(A, B) = \max \left\{ \sup_{a \in A} \inf_{b \in B} \|a - b\|, \sup_{b \in B} \inf_{a \in A} \|a - b\| \right\} = \max \left\{ |a^L - b^L|, |a^U - b^U| \right\}$$

donde $\|\bullet\|$ es la norma Euclidiana.

Definición 1.2.6. (Distancia Hausdorff Infinito): Sean A y B dos elementos del conjunto $I^m(\mathbb{R})$. Se define la métrica Hausdorff infinito $d_{H_\infty}(A, B)$ como:

$$d_{H_\infty}(A, B) = \max_{i \in 1, \dots, m} \{d_H(A_i, B_i)\}$$

Definición 1.2.7. (Límite Función Multi-intervalo-valuada): Sea $f(x)$ una función multi-intervalo-valuada definida en \mathbb{R}^n y $A \in I^m(\mathbb{R})$. Para $c \in \mathbb{R}^n$ se dice que

$$\lim_{x \rightarrow c} f(x) = A \text{ si y sólo si } \forall \epsilon > 0 \exists \delta > 0 : \|x - c\| < \delta \rightarrow d_{H_\infty}(f(x), A) < \epsilon.$$

También se puede observar que $\lim_{x \rightarrow c} f(x) = A \leftrightarrow \lim_{x \rightarrow c} f_i^L(x) = a_i^L \wedge \lim_{x \rightarrow c} f_i^U(x) = a_i^U$ para todo $i = 1, \dots, m$

Definición 1.2.8. (Continuidad): Sea $f(x)$ una función multi intervalo-valuada definida en \mathbb{R}^n . Se dice que $f(x)$ es continua en $c \in \mathbb{R}^n$ si $\lim_{x \rightarrow c} f(x) = f(c)$

1.2.3. Diferenciabilidad en Funciones Multi-intervalo-valuadas

En esta sección se definirán dos tipos de diferenciabilidad, las cuales son aplicadas al momento de definir el concepto de solución para los problemas de optimización multi-objetivo intervalo-valuados.

Definición 1.2.9. (Función débilmente diferenciable): Sea X un subconjunto abierto de \mathbb{R} . Una función $f(x) = [f^L(x), f^U(x)]$ intervalo-valuada definida en X es llamada débilmente diferenciable en x_0 si las funciones real-valuadas $f^L(x)$ y $f^U(x)$ son diferenciables en x_0 .

Definición 1.2.10. (Función H -diferenciable): Sea X un subconjunto abierto de \mathbb{R} . Una función $f(x) = [f^L(x), f^U(x)]$ intervalo-valuada definida en X es llamada H -diferenciable (fuertemente diferenciable) en x_0 si existe un intervalo cerrado $A(x_0)$ (el intervalo depende de x_0) en \mathbb{R} tal que $\lim_{h \rightarrow 0^+} \frac{f(x_0+h) \ominus f(x_0)}{h}$ y $\lim_{h \rightarrow 0^+} \frac{f(x_0) \ominus f(x_0-h)}{h}$ existan y sean iguales a $A(x_0)$. En este caso $A(x_0)$ es llamado H -derivada de f en x_0 .

Proposición 1.2.2. : Sea f una función intervalo-valuada definida en $X \subset \mathbb{R}^n$.

- Si f es H -diferenciable en $x_0 \in X$, entonces f es débilmente diferenciable en x_0 .
- Si f es H -diferenciable continuamente en $x_0 \in X$, entonces f es débilmente diferenciable continuamente en x_0 .

1.2.4. Problema de Optimización y Concepto de Solución

Definición 1.2.11. (Problema de Optimización Multi-intervalo-valuado): Un problema de optimización multi-intervalo-valuado se define como:

$$(MIVP1) \min f(x) : \mathbb{R}^n \rightarrow I^m(\mathbb{R}) \quad (1.1)$$

$$s.a. x \in X \subset \mathbb{R}^n$$

donde $f(x) = \begin{bmatrix} f_1(x) \\ \dots \\ f_m(x) \end{bmatrix}$ y cada $f_k(x) = f_k(x_1, \dots, x_n) = [f_k^L(x_1, \dots, x_n), f_k^U(x_1, \dots, x_n)] = [f_k^L(x), f_k^U(x)]$ es una función intervalo-valuada para $k = 1, \dots, m$ y el conjunto factible X es asumido como un conjunto **convexo** en \mathbb{R}^n

Esta es la forma general de presentar el problema de optimización multi-objetivo intervalo-valuado, donde el conjunto factible puede ser expresado a través de restricciones real-valuadas o intervalo-valuadas.

La representación cuando el problema tiene restricciones real-valuadas es la siguiente:

$$(MIVP2) \min f(x) : \mathbb{R}^n \rightarrow I^m(\mathbb{R}) \quad (1.2)$$

$$s.a. g_i(x) \leq 0, i = 1, \dots, p$$

donde las funciones real-valuadas $g_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ convexas en \mathbb{R}^n , para $i = 1, \dots, p$. En este problema el conjunto factible esta definido como $X = \{x : g_i(x) \leq 0, i = 1, \dots, p\}$

Cuando las restricciones son intervalo-valuadas se formula de la siguiente forma

$$(MIVP3) \min f(x) : \mathbb{R}^n \rightarrow I^m(\mathbb{R}) \quad (1.3)$$

$$s.a. G_i(x) \leq [0, 0], i = 1, \dots, p$$

donde las funciones intervalo-valuadas $G_i(x) = [G_i^L(x), G_i^U(x)]$ son funciones intervalo-valuadas definidas en \mathbb{R}^n , para $i = 1, \dots, p$. Ahora es problema puede formularse de la siguiente forma:

$$(MIVP4) \min f(x) : \mathbb{R}^n \rightarrow I^m(\mathbb{R}) \quad (1.4)$$

$$s.a. G_i^L(x) \leq 0, i = 1, \dots, p$$

$$G_i^U(x) \leq 0, i = 1, \dots, p$$

Se puede ver que el conjunto factible para los problemas (MIVP3) y (MIVP4) son idénticas. Además, este último es idéntico al problema (MIVP2).

Ahora, sea x^* la solución factible del problema (MIVP1), entonces $f(x^*)$ es un vector intervalo-valorado, es decir $f(x^*) \in I^m(\mathbb{R})$. A continuación se definirán los conceptos de órdenes y dominancia de Pareto, los cuales son necesarios para luego definir los conceptos de solución para este tipo de problemas.

Definición 1.2.12. (*Órdenes Parciales en $I(\mathbb{R})$*): Sea A un intervalo cerrado en \mathbb{R} . Se define el centro del intervalo como $a_c = \frac{a^U + a^L}{2}$ y la longitud media como $a_w = \frac{a^U - a^L}{2}$. Se definen los órdenes parciales para A y $B \in I(\mathbb{R})$ como:

1. $A \leq_{LU} B \leftrightarrow a^L \leq b^L \wedge a^U \leq b^U$
2. $A \leq_{CW} B \leftrightarrow a_c \leq b_c \wedge a_w \leq b_w$
3. $A <_{LU} B \leftrightarrow A \leq_{LU} B \wedge A \neq B$
4. $A <_{CW} B \leftrightarrow A \leq_{CW} B \wedge A \neq B$
5. $A \leq_{UC} B \leftrightarrow a^U \leq b^U \wedge a_c \leq b_c$
6. $A <_{UC} B \leftrightarrow A \leq_{UC} B \wedge A \neq B$

Definición 1.2.13. (*Órdenes Parciales en $I^m(\mathbb{R})$*): Sea A y $B \in I^m(\mathbb{R})$, se definen los órdenes parciales como:

1. $A \leq_{LU} B \leftrightarrow A_k \leq_{LU} B_k$ para $k = 1, \dots, m$
2. $A <_{LU} B \leftrightarrow A_k \leq_{LU} B_k$ para $k = 1, \dots, m$ y $A_i <_{LU} B_i$ para al menos un i
3. $A \leq_{CW} B \leftrightarrow A_k \leq_{CW} B_k$ para $k = 1, \dots, m$
4. $A <_{CW} B \leftrightarrow A_k \leq_{CW} B_k$ para $k = 1, \dots, m$ y $A_i <_{CW} B_i$ para al menos un i

Definición 1.2.14. (*Dominancia de Pareto*): Un punto $x = (x_1, \dots, x_n)$ se dice que domina a otro punto $x^* = (x_1^*, \dots, x_n^*)$ si y sólo si $\forall j \in 1, \dots, m, f_j(x) \leq f_j(x^*) \wedge \exists j \in 1, \dots, m, f_j(x) < f_j(x^*)$. En este caso se escribe $x > x^*$.

Definición 1.2.15. (*Punto Óptimo de Pareto*): Un punto $x = (x_1, \dots, x_n) \in X$ se dice que es óptimo si y solo si $\neg \exists x^* \in X, x^* > x$.

Definición 1.2.16. (*Solución Factible Tipo I*): Sea x^* una solución factible del problema (MIVP1):

- Se dice que x^* es una solución óptima de Pareto tipo I del problema (MIVP1) si no existe $\bar{x} \in X$ tal que $f(\bar{x}) <_{LU} f(x^*)$.
- Se dice que x^* es una solución óptima de Pareto tipo I fuerte del problema (MIVP1) si no existe $\bar{x} \in X$ tal que $f(\bar{x}) \leq_{LU} f(x^*)$.
- Se dice que x^* es una solución óptima de Pareto tipo I débil del problema (MIVP1) si no existe $\bar{x} \in X$ tal que $f_k(\bar{x}) <_{LU} f_k(x^*)$ para todo $k = 1, \dots, m$.

Note que si $X_{wp}^{(I)}, X_p^{(I)}$ y $X_{sp}^{(I)}$ denotan las soluciones óptima de Pareto tipo I débil, óptima de Pareto tipo I y óptima de Pareto tipo I fuerte respectivamente, entonces $X_{sp}^{(I)} \subset X_p^{(I)} \subset X_{wp}^{(I)}$

Definición 1.2.17. (Solución Factible Tipo II): Sea x^* una solución factible del problema (MIVP1):

- Se dice que x^* es una solución óptima de Pareto tipo II del problema (MIVP1) si no existe $\bar{x} \in X$ tal que $f(\bar{x}) <_{CW} f(x^*)$
- Se dice que x^* es una solución óptima de Pareto tipo II fuerte del problema (MIVP1) si no existe $\bar{x} \in X$ tal que $f(\bar{x}) \leq_{CW} f(x^*)$
- Se dice que x^* es una solución óptima de Pareto tipo II débil del problema (MIVP1) si no existe $\bar{x} \in X$ tal que $f_k(\bar{x}) <_{CW} f_k(x^*)$ para todo $k = 1, \dots, m$

Note que si $X_{wp}^{(II)}, X_p^{(II)}$ y $X_{sp}^{(II)}$ denotan las soluciones óptima de Pareto tipo II débil, óptima de Pareto tipo II y óptima de Pareto tipo II fuerte respectivamente, entonces $X_{sp}^{(II)} \subset X_p^{(II)} \subset X_{wp}^{(II)}$

1.2.5. Condiciones Karush-Kuhn-Tucker (KKT) de Optimalidad

En lo que sigue considérese el problema de optimización multi-objetivo con funciones intervalo-valoradas (MIVP2).

1.2.5.1. Condiciones KKT para el caso continuo débilmente diferenciable

En esta sección se considerará la función intervalo-valorada $f_k, k = 1, \dots, m$, continua débilmente diferenciable en alguna solución factible x^*

Definición 1.2.18. : Sea X un subconjunto no vacío de \mathbb{R}^n y f una función intervalo-valorada en X . Se dice que f es LU-convexa en x^* si $f(\lambda x^* + (1 - \lambda)x) \leq_{LU} \lambda f(x^*) + (1 - \lambda)f(x)$ para todo $x \in X$ y $\lambda \in (0, 1)$. Se define similarmente la CW-convexidad en x^* con la relación de orden \leq_{CW} .

Proposición 1.2.3. : Sea X un subconjunto no vacío de \mathbb{R}^n y f una función intervalo-valorada en X . Las siguientes sentencias son verdaderas:

- f es LU-convexa en x^* si y sólo si f^L y f^U son convexas en x^*
- f es CW-convexa en x^* si y sólo si f^C y f^W son convexas en x^*

A continuación se presentarán las condiciones KKT para el problema (MIVP2)

Primero se presentarán las condiciones KKT para las soluciones óptimas de Pareto de tipo I y tipo II

Teorema 1.2.1. : Sean las funciones real-valoradas de restricción $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ convexas en \mathbb{R}^n y continuamente diferenciables en x^* para $i : 1, \dots, p$. Sea $X = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \dots, p\}$ el conjunto factible y $x^* \in X$. Sea $f_k : \mathbb{R}^n \rightarrow I(\mathbb{R})$ continua débilmente diferenciable en x^* para $k = 1, \dots, m$. Las siguientes afirmaciones son verdaderas:

- Suponga que las funciones objetivo intervalo-valuadas $f_k : \mathbb{R}^n \rightarrow I(\mathbb{R})$ son LU-convexas en x^* para $k = 1, \dots, m$. Si existen multiplicadores de Lagrange $0 < \lambda_k^L, \lambda_k^U \in \mathbb{R}$, $k = 1, \dots, m$ y $0 \leq \mu_i \in \mathbb{R}$, $i = 1, \dots, p$, tal que:

- $\sum_{k=1}^m \lambda_k^L \nabla f_k^L(x^*) + \sum_{k=1}^m \lambda_k^U \nabla f_k^U(x^*) + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$
- $\mu_i g_i(x^*) = 0$, para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto tipo I del problema (MIVP2)

- Suponga que las funciones objetivo intervalo-valuadas $f_k : \mathbb{R}^n \rightarrow I(\mathbb{R})$ son CW-convexas en x^* para $k = 1, \dots, m$. Si existen multiplicadores de Lagrange $0 < \lambda_k^C, \lambda_k^W \in \mathbb{R}$, $k = 1, \dots, m$ y $0 \leq \mu_i \in \mathbb{R}$, $i = 1, \dots, p$, tal que:

- $\sum_{k=1}^m \lambda_k^C \nabla f_k^C(x^*) + \sum_{k=1}^m \lambda_k^W \nabla f_k^W(x^*) + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$
- $\mu_i g_i(x^*) = 0$, para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto tipo II del problema (MIVP2)

- Suponga que las funciones objetivo intervalo-valuadas $f_k : \mathbb{R}^n \rightarrow I(\mathbb{R})$ son CW-convexas en x^* para $k = 1, \dots, m$. Si existen multiplicadores de Lagrange $0 < \lambda_k^L, \lambda_k^U \in \mathbb{R}$, $k = 1, \dots, m$ y $0 \leq \mu_i \in \mathbb{R}$, $i = 1, \dots, p$, tal que:

- $\lambda_k^L < \lambda_k^U$, para todo $k = 1, \dots, m$
- $\sum_{k=1}^m \lambda_k^L \nabla f_k^L(x^*) + \sum_{k=1}^m \lambda_k^U \nabla f_k^U(x^*) + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$
- $\mu_i g_i(x^*) = 0$, para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto tipo II del problema (MIVP2)

La prueba de este teorema se puede encontrar en [31]

Definición 1.2.19. : Sea $f(x) = [f^L(x), f^U(x)]$ una función intervalo-valuada definida en el conjunto convexo no vacío $X \in \mathbb{R}^n$. Se dice f es LU-seudoconvexa (estrictamente LU-seudoconvexa) en x^* si y sólo si f^L y f^U son seudoconvexas (estrictamente seudoconvexas) en x^* . Se dice que f es CW-seudoconvexa (estrictamente CW-seudoconvexa) en x^* si y sólo si f^C y f^W son seudoconvexas (estrictamente seudoconvexas) en x^* .

Teorema 1.2.2. : Sean las funciones real-valuadas de restricción $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ convexas en \mathbb{R}^n y continuamente diferenciables en x^* para $i = 1, \dots, p$. Sea $X = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \dots, p\}$ el conjunto factible y $x^* \in X$. Sea $f_k : \mathbb{R}^n \rightarrow I(\mathbb{R})$ continua débilmente diferenciable en x^* para $k = 1, \dots, m$. Las siguientes afirmaciones son verdaderas:

- Suponga que las funciones objetivo intervalo-valuadas $f_k : \mathbb{R}^n \rightarrow I(\mathbb{R})$ son LU-seudoconvexas o LU-convexas en x^* para $k = 1, \dots, m$. Si existen multiplicadores de Lagrange $0 < \lambda_k^L, \lambda_k^U \in \mathbb{R}$, $k = 1, \dots, m$ y $0 \leq \mu_i^L, \mu_i^U \in \mathbb{R}$, $i = 1, \dots, p$, tal que:

- $\sum_{k=1}^m \lambda_k^L \nabla f_k^L(x^*) + \sum_{i=1}^p \mu_i^L \nabla g_i(x^*) = 0$
- $\sum_{k=1}^m \lambda_k^U \nabla f_k^U(x^*) + \sum_{i=1}^p \mu_i^U \nabla g_i(x^*) = 0$
- $\mu_i^L g_i(x^*) = \mu_i^U g_i(x^*) = 0$, para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto tipo I del problema (MIVP2)

- Suponga que las funciones objetivo intervalo-valuadas $f_k : \mathbb{R}^n \rightarrow I(\mathbb{R})$ son CW-seudoconvexas o CW-convexas en x^* para $k = 1, \dots, m$. Si existen multiplicadores de Lagrange $0 < \lambda_k^C, \lambda_k^W \in \mathbb{R}$, $k = 1, \dots, m$ y $0 \leq \mu_i^C, \mu_i^W \in \mathbb{R}$, $i = 1, \dots, p$, tal que:

- $\sum_{k=1}^m \lambda_k^C \nabla f_k^C(x^*) + \sum_{i=1}^p \mu_i^C \nabla g_i(x^*) = 0$
- $\sum_{k=1}^m \lambda_k^C \nabla f_k^W(x^*) + \sum_{i=1}^p \mu_i^W \nabla g_i(x^*) = 0$
- $\mu_i^C g_i(x^*) = \mu_i^W g_i(x^*) = 0$, para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto tipo II del problema (MIVP2)

La prueba se este teorema se puede encontrar en [31]

Teorema 1.2.3. : Sean las funciones real-valuadas de restricción $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ convexas en \mathbb{R}^n y continuamente diferenciables en x^* para $i = 1, \dots, p$. Sea $X = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \dots, p\}$ el conjunto factible y $x^* \in X$. Sea $f_k : \mathbb{R}^n \rightarrow I(\mathbb{R})$ continua débilmente diferenciable en x^* para $k = 1, \dots, m$. Las siguientes afirmaciones son verdaderas:

- Suponga que las funciones objetivo intervalo-valuadas $f_k : \mathbb{R}^n \rightarrow I(\mathbb{R})$ son todas LU-seudoconvexas en x^* para $k = 1, \dots, m$. Si para todo $k \in \{1, \dots, m\}$ existen multiplicadores de Lagrange $0 \leq \mu_{ki}^L, \mu_{ki}^U \in \mathbb{R}$, $i = 1, \dots, p$, tal que:

- $\nabla f_k^L(x^*) + \sum_{i=1}^p \mu_{ki}^L \nabla g_i(x^*) = 0$
- $\nabla f_k^U(x^*) + \sum_{i=1}^p \mu_{ki}^U \nabla g_i(x^*) = 0$
- $\mu_{ki}^L g_i(x^*) = \mu_{ki}^U g_i(x^*) = 0$, para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto tipo I del problema (MIVP2)

- Suponga que las funciones objetivo intervalo-valuadas $f_k : \mathbb{R}^n \rightarrow I(\mathbb{R})$ son todas CW-seudoconvexas en x^* para $k = 1, \dots, m$. Si para todo $k \in \{1, \dots, m\}$ existen multiplicadores de Lagrange $0 \leq \mu_{ki}^C, \mu_{ki}^W \in \mathbb{R}$, $i = 1, \dots, p$, tal que:

- $\nabla f_k^C(x^*) + \sum_{i=1}^p \mu_{ki}^C \nabla g_i(x^*) = 0$
- $\nabla f_k^W(x^*) + \sum_{i=1}^p \mu_{ki}^W \nabla g_i(x^*) = 0$
- $\mu_{ki}^C g_i(x^*) = \mu_{ki}^W g_i(x^*) = 0$, para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto tipo II del problema (MIVP2)

La prueba se este teorema se puede encontrar en [31]

A continuación se presentarán las condiciones KKT para las soluciones óptimas de Pareto de tipo I y tipo II débiles

Teorema 1.2.4. : Sean las funciones real-valuadas de restricción $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ convexas en \mathbb{R}^n y continuamente diferenciables en x^* para $i = 1, \dots, p$. Sea $X = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \dots, p\}$ el conjunto factible y $x^* \in X$. Las siguientes afirmaciones son verdaderas:

- Suponga que existe una función objetivo intervalo-valuada $f_h(x) : \mathbb{R}^n \rightarrow I(\mathbb{R})$ que es LU-convexa y continua débilmente diferenciable en x^* . Si existen multiplicadores de Lagrange $0 < \lambda^L, \lambda^U \in \mathbb{R}$ y $0 \leq \mu_i \in \mathbb{R}$, $i = 1, \dots, p$ tales que:

- $\lambda^L \nabla f_h^L + \lambda^U \nabla f_h^U + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$
- $\mu_i g_i(x^*) = 0$ para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto tipo I débil del problema (MIVP2)

- Suponga que existe una función objetivo intervalo-valuada $f_h(x) : \mathbb{R}^n \rightarrow I(\mathbb{R})$ que es CW-convexa y continua débilmente diferenciable en x^* . Si existen multiplicadores de Lagrange $0 < \lambda^C, \lambda^W \in \mathbb{R}$ y $0 \leq \mu_i \in \mathbb{R}$, $i = 1, \dots, p$ tales que:

- $\lambda^C \nabla f_h^C + \lambda^W \nabla f_h^W + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$
- $\mu_i g_i(x^*) = 0$ para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto tipo II débil del problema (MIVP2)

- Suponga que existe una función objetivo intervalo-valuada $f_h(x) : \mathbb{R}^n \rightarrow I(\mathbb{R})$ que es CW-convexa y continua débilmente diferenciable en x^* . Si existen multiplicadores de Lagrange $0 < \lambda^L, \lambda^U \in \mathbb{R}$ y $0 \leq \mu_i \in \mathbb{R}$, $i = 1, \dots, p$ tales que:

- $\lambda^L < \lambda^U$
- $\lambda^L \nabla f_h^L + \lambda^U \nabla f_h^U + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$
- $\mu_i g_i(x^*) = 0$ para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto tipo II débil del problema (MIVP2)

La prueba se este teorema se puede encontrar en [31]

Teorema 1.2.5. : Sean las funciones real-valuadas de restricción $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ convexas en \mathbb{R}^n y continuamente diferenciables en x^* para $i = 1, \dots, p$. Sea $X = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \dots, p\}$ el conjunto factible y $x^* \in X$. Las siguientes afirmaciones son verdaderas:

- Suponga que existe una función objetivo intervalo-valuada $f_h(x) : \mathbb{R}^n \rightarrow I(\mathbb{R})$ que es LU-convexa o LU-seudoconvexa y continua débilmente diferenciable en x^* . Si existen multiplicadores de Lagrange $0 \leq \mu_i^L, \mu_i^U \in \mathbb{R}$, para $i = 1, \dots, p$ tales que:

- $\nabla f_h^L + \sum_{i=1}^p \mu_i^L \nabla g_i(x^*) = 0$
- $\nabla f_h^U + \sum_{i=1}^p \mu_i^U \nabla g_i(x^*) = 0$
- $\mu_i^L g_i(x^*) = \mu_i^U g_i(x^*) = 0$ para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto tipo I débil del problema (MIVP2)

- Suponga que existe una función objetivo intervalo-valuada $f_h(x) : \mathbb{R}^n \rightarrow I(\mathbb{R})$ que es CW-convexa o CW-seudoconvexa y continua débilmente diferenciable en x^* . Si existen multiplicadores de Lagrange $0 \leq \mu_i^C, \mu_i^W \in \mathbb{R}$, para $i = 1, \dots, p$ tales que:

- $\nabla f_h^C + \sum_{i=1}^p \mu_i^C \nabla g_i(x^*) = 0$

- $\nabla f_h^W + \sum_{i=1}^p \mu_i^W \nabla g_i(x^*) = 0$
- $\mu_i^C g_i(x^*) = \mu_i^W g_i(x^*) = 0$ para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto tipo II débil del problema (MIVP2)

La prueba se este teorema se puede encontrar en [31]

Por último se presentarán las condiciones KKT para las soluciones óptimas de Pareto de tipo I y tipo II fuertes

Definición 1.2.20. : Sea $f(x) = [f^L(x), f^U(x)] = \langle f^C(x), f^W(x) \rangle$ una función intervalo-valuada definida sobre un subconjunto convexo no vacío $X \subset \mathbb{R}^n$. Se dice que f es estrictamente L-seudoconvexa en x^* si y sólo si f^L es estrictamente seudoconvexa en x^* . Similarmente se define estrictamente U-seudoconvexa, estrictamente C-seudoconvexa y estrictamente W-seudoconvexa para f^U , f^C y f^W respectivamente.

Note que f es estrictamente LU-seudoconvexa en x^* si y sólo si es estrictamente L- y U-seudoconvexa en x^* y estrictamente CW-seudoconvexa en x^* si y sólo si es estrictamente C- y W-seudoconvexa en x^* .

Teorema 1.2.6. : Sean las funciones real-valuadas de restricción $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ convexas en \mathbb{R}^n y continuamente diferenciables en x^* para $i : 1, \dots, p$. Sea $X = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \dots, p\}$ el conjunto factible y $x^* \in X$. Las siguientes afirmaciones son verdaderas:

- Suponga que existe una función objetivo intervalo-valuada $f_h(x) : \mathbb{R}^n \rightarrow I(\mathbb{R})$ que es estrictamente L-seudoconvexa (respectivamente estrictamente U-seudoconvexa) y continua débilmente diferenciable en x^* . Si existen multiplicadores de Lagrange $0 \leq \mu_i \in \mathbb{R}, i = 1, \dots, p$ tales que:

- $\nabla f_h^L + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$ (respectivamente $\nabla f_h^U + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$)
- $\mu_i g_i(x^*) = 0$ para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto tipo I fuerte del problema (MIVP2)

- Suponga que existe una función objetivo intervalo-valuada $f_h(x) : \mathbb{R}^n \rightarrow I(\mathbb{R})$ que es estrictamente C-seudoconvexa (respectivamente estrictamente W-seudoconvexa) y continua débilmente diferenciable en x^* . Si existen multiplicadores de Lagrange $0 \leq \mu_i \in \mathbb{R}, i = 1, \dots, p$ tales que:

- $\nabla f_h^C + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$ (respectivamente $\nabla f_h^W + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$)
- $\mu_i g_i(x^*) = 0$ para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto tipo II fuerte del problema (MIVP2)

La prueba se este teorema se puede encontrar en [31]

1.2.5.2. Condiciones KKT para el caso continuo H-diferenciable

Para empezar se presentarán las condiciones KKT para la forma intervalo-valuada. Sea $f(x) = [f^L(x), f^U(x)]$ una función intervalo-valuada definida en \mathbb{R}^n y H-diferenciable en $x_0 \in \mathbb{R}^n$. El gradiente

de f en x_0 , denotado por $\nabla f(x_0)$, es definido por

$$\nabla f(x_0) = \left(\frac{\partial f}{\partial x_1}(x_0), \dots, \frac{\partial f}{\partial x_n}(x_0) \right)^T$$

donde $\frac{\partial f}{\partial x_j}(x_0)$ es la H-derivada parcial j de f en x_0 , es decir

$$\frac{\partial f}{\partial x_j}(x_0) = \left[\frac{\partial f^L}{\partial x_j}(x_0), \frac{\partial f^U}{\partial x_j}(x_0) \right] \text{ un intervalo cerrado.}$$

Sea g_i una función real valuada definida en \mathbb{R}^n para $i = 1, \dots, p$, entonces

$$\nabla g_i(x_0) = \left(\frac{\partial g_i}{\partial x_1}(x_0), \dots, \frac{\partial g_i}{\partial x_n}(x_0) \right)^T$$

donde $\frac{\partial g_i}{\partial x_j}(x_0)$ puede ser expresada como el intervalo cerrado

$$\left[\frac{\partial g_i}{\partial x_j}(x_0), \frac{\partial g_i}{\partial x_j}(x_0) \right]$$

Entonces la siguiente ecuación:

$$\sum_{k=1}^m \lambda_k \nabla f_k(x_0) + \sum_{i=1}^p \mu_i \nabla g_i(x_0) = 0 \quad (1.5)$$

donde $0 < \lambda_k \in \mathbb{R}$ para $k = 1, \dots, m$ y $0 \leq \mu_i \in \mathbb{R}$ para $i = 1, \dots, p$ puede ser expresada como:

$$\sum_{k=1}^m \lambda_k \frac{\partial f_k^L}{\partial x_j}(x_0) + \sum_{i=1}^p \mu_i \frac{\partial g_i}{\partial x_j}(x_0) = 0 = \sum_{k=1}^m \lambda_k \frac{\partial f_k^U}{\partial x_j}(x_0) + \sum_{i=1}^p \mu_i \frac{\partial g_i}{\partial x_j}(x_0) \quad (1.6)$$

para todo $j = 1, \dots, n$. También se puede escribir como

$$\sum_{k=1}^m \lambda_k \nabla f_k^L(x_0) + \sum_{i=1}^p \mu_i \nabla g_i(x_0) = 0 = \sum_{k=1}^m \lambda_k \nabla f_k^U(x_0) + \sum_{i=1}^p \mu_i \nabla g_i(x_0) \quad (1.7)$$

lo cual implica

$$\sum_{k=1}^m \lambda_k \nabla f_k^L(x_0) + \sum_{k=1}^m \lambda_k \nabla f_k^U(x_0) + \sum_{i=1}^p \bar{\mu}_i \nabla g_i(x_0) = 0 \quad (1.8)$$

donde $\bar{\mu}_i = 2\mu_i$ para todo $i = 1, \dots, p$

Teorema 1.2.7. : Sean las funciones real-valuadas de restricción $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ convexas en \mathbb{R}^n y continuamente diferenciables en x^* para $i = 1, \dots, p$. Sea $X = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \dots, p\}$ el conjunto factible y $x^* \in X$. Suponga que las funciones objetivo intervalo-valuadas $f_k : \mathbb{R}^n \rightarrow I(\mathbb{R})$ son LU-convexas y H-diferenciables en x^* para $k = 1, \dots, m$. Si existen multiplicadores de Lagrange $0 < \lambda_k \in \mathbb{R}$, $k = 1, \dots, m$ y $0 \leq \mu_i \in \mathbb{R}$, $i = 1, \dots, p$ tales que:

- $\sum_{k=1}^m \lambda_k \nabla f_k(x^*) + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$
- $\mu_i g_i(x^*) = 0$ para todo $i = 1, \dots, p$

entonces x^* es solución óptima de Pareto de tipo I del problema (MIVP2)

La prueba de este teorema se puede encontrar en [31]

Definición 1.2.21. : Sea $f(x) = [f^L(x), f^U(x)]$ una función intervalo-valuada sobre un subconjunto convexo $X \subset \mathbb{R}^n$

- Se dice que f es LU-nocreciente (respectivamente CW-nocreciente) en x^* si $x \geq x^*$ si y sólo si $f(x) \leq_{LU} f(x^*)$ (respectivamente $f(x) \leq_{CW} f(x^*)$) para todo $x \in X$
- Se dice que f es LU-decreciente (respectivamente CW-decreciente) en x^* si $x > x^*$ si y sólo si $f(x) <_{LU} f(x^*)$ (respectivamente $f(x) <_{CW} f(x^*)$) para todo $x \in X$

Teorema 1.2.8. : Sean las funciones real-valuadas de restricción $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ convexas en \mathbb{R}^n y continuamente diferenciables en x^* para $i : 1, \dots, p$. Sea $X = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \dots, p\}$ el conjunto factible y $x^* \in X$. Las siguientes afirmaciones son verdaderas:

- Suponga que existe una función objetivo intervalo-valuada $f_h(x) : \mathbb{R}^n \rightarrow I(\mathbb{R})$ tal que es uno-a-uno y LU-decreciente, a la vez es U-convexa y H-diferenciable en x^* . Además, asuma que $\nabla f_h^L(x^*) \neq \nabla f_h^U(x^*)$. Si existen multiplicadores de Lagrange $0 \leq \mu_i \in \mathbb{R}, i = 1, \dots, p$ tales que:

- $\nabla f_h^L(x^*) + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$
- $\mu_i g_i(x^*) = 0$ para todo $i = 1, \dots, p$

entonces x^* es una solución óptima de Pareto tipo I fuerte del problema (MIVP2)

- Suponga que existe una función objetivo intervalo-valuada $f_h(x) : \mathbb{R}^n \rightarrow I(\mathbb{R})$ tal que es uno-a-uno y CW-decreciente, a la vez es C-convexa (respectivamente W-convexa) y H-diferenciable en x^* . Además, asumase que $\nabla f_h^L(x^*) > 0$ (respectivamente $\nabla f_h^L(x^*) < 0$). Si existen multiplicadores de Lagrange $0 \leq \mu_i \in \mathbb{R}, i = 1, \dots, p$ tales que:

- $\nabla f_h^W(x^*) + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$ (respectivamente $\nabla f_h^C(x^*) + \sum_{i=1}^p \mu_i \nabla g_i(x^*) = 0$)
- $\mu_i g_i(x^*) = 0$ para todo $i = 1, \dots, p$

entonces x^* es una solución óptima de Pareto tipo II fuerte del problema (MIVP2)

La prueba de este teorema se puede encontrar en [31]

1.2.5.3. Ejemplos Numéricos

A continuación se presentarán diferentes tipos de problemas de optimización bajo incertidumbre:

Ejemplo 1.2.1. Considere el siguiente problema de optimización intervalo-valuado:

$$\begin{aligned} \min f(x) &= [x^2 + x + 1, x^2 + 3] \\ \text{s.a. } x - 2 &\leq 0 \\ -x &\leq 0 \end{aligned} \tag{1.9}$$

Se tiene que las funciones objetivos son $f^L(x) = x^2 + x + 1$, $f^U(x) = x^2 + 3$ y las restricciones son $g_1(x) = x - 2$, $g_2(x) = -x$. Se puede observar que las funciones cumplen con las condiciones de KKT. Por lo tanto aplicando el teorema se obtiene:

$$\lambda^L * (2x + 1) + \lambda^U * 2x + \mu_1 - \mu_2 = 0$$

$\mu_1 * (x - 2) = 0 = \mu_2$ Si $x^* = 0$, entonces por la segunda condición se obtiene que $\mu_1 = 0$, por lo tanto reemplazando en la primer condición se tiene que $\lambda^L = \mu_2$. Tomando los multiplicadores $\lambda^L = \lambda^U = \mu_2 = 1$ y $\mu_1 = 0$, aplicando el teorema se tiene que $x^* = 0$ es solución de tipo I y tipo II.

Ejemplo 1.2.2. Considere el siguiente problema de optimización multi-objetivo intervalo-valuado

$$\begin{aligned} \min f(x_1, x_2) &= \left[\begin{array}{c} [x_1^2 + x_2^2 + 1, x_1^2 + x_2^2 + 2] \\ [2x_1^2 + 2x_2^2 + 3, 2x_1^2 + 2x_2^2 + 4] \end{array} \right] \\ \text{s.a. } x_1 + x_2 &\geq 1 \\ 6x_1 + 2x_2 &\geq 12 \\ x_1 \geq 0 \quad x_2 &\geq 0 \end{aligned} \quad (1.10)$$

Entonces se tienen las siguientes funciones objetivo

$$f_1^L = x_1^2 + x_2^2 + 1, f_1^U = x_1^2 + x_2^2 + 2, f_2^L = 2x_1^2 + 2x_2^2 + 3 \text{ y } f_2^U = 2x_1^2 + 2x_2^2 + 4$$

y las siguientes restricciones

$$g_1(x) = 1 - x_1 - x_2, g_2(x) = 12 - 6x_1 - 2x_2, g_3(x) = -x_1 \text{ y } g_4(x) = -x_2$$

Aplicando el teorema se tiene:

$$\lambda_1^L \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} + \lambda_2^L \begin{bmatrix} 4x_1 \\ 4x_2 \end{bmatrix} + \lambda_1^U \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} + \lambda_2^U \begin{bmatrix} 4x_1 \\ 4x_2 \end{bmatrix} + \mu_1 \begin{bmatrix} -1 \\ -1 \end{bmatrix} + \mu_2 \begin{bmatrix} -6 \\ -2 \end{bmatrix} + \mu_3 \begin{bmatrix} -1 \\ 0 \end{bmatrix} + \mu_4 \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

es decir, se debe resolver el siguiente sistema de ecuaciones:

$$2\lambda_1^L x_1 + 4\lambda_2^L x_1 + 2\lambda_1^U x_1 + 4\lambda_2^U x_1 - \mu_1 - 6\mu_2 - \mu_3 = 0$$

$$2\lambda_1^L x_2 + 4\lambda_2^L x_2 + 2\lambda_1^U x_2 + 4\lambda_2^U x_2 - \mu_1 - 2\mu_2 - \mu_4 = 0$$

resolviéndolo algebraicamente se obtiene:

$$(x_1^*, x_2^*) = \left(\frac{9}{5}, \frac{3}{5}\right)$$

$$\lambda_1^L = \lambda_1^U = \frac{1}{2} \text{ y } \lambda_2^L = \lambda_2^U = \frac{1}{4}$$

$$\mu_1 = \mu_3 = \mu_4 = 0 \text{ y } \mu_2 = \frac{6}{5}$$

como $g_2\left(\frac{9}{5}, \frac{3}{5}\right) = 0$, entonces $\mu_i g_i(x) = 0$ para $i = 1, \dots, 4$. Entonces se concluye $(x_1^*, x_2^*) = \left(\frac{9}{5}, \frac{3}{5}\right)$ es solución óptima de Pareto tipo I.

Ejemplo 1.2.3. Considere el problema de cercha presentado en la figura 1.1. El objetivo es optimizar el área de las barras para obtener el mínimo desplazamiento en la unión 2 en la dirección y. Las barras 1, 2, 3, y 4 tiene la misma área seccional, denotada por A_1 . Las barras 7 y 8, 9 y 10, 5 y 6 tiene la misma área y son denotadas por A_2 , A_3 y A_4 respectivamente. Las cargas p_1 y p_3 y la densidad ρ son 1000N, 1000N y $7,8 \times 10^{-3} \text{ kg/mm}^3$, respectivamente. La carga p_2 y el modulo de Young E son inciertos, tienen como valor medio 1000N y $2,0 \times 10^5 \text{ Mp}$ con un nivel de incertidumbre del 10 % con respecto al valor medio, para ambos. El máximo valor del peso que soporta la cercha es 1365kg [11].

Este problema se puede formular de la siguiente forma:

$$\begin{aligned} \min u_{2y}(A, E, p_2) \\ \text{s.a. } w &= (4000A_1 + 2000\sqrt{2}A_2 + 2000\sqrt{2}A_3 + 2000A_4) \\ A_i &\in [5, 25], \quad i = 1, 2, 3, 4 \\ p_2 &\in [900, 1100] \\ E &\in [1,8 \times 10^5, 2,2 \times 10^5] \end{aligned} \quad (1.11)$$

Como puede observarse este problema ya no es simple resolver analíticamente, la función objetivo no es una función lineal, por lo tanto verificar las condiciones de diferenciabilidad y convexidad es un poco más complejo. Cuando se tienen problemas más complejos o que no satisfacen alguna de

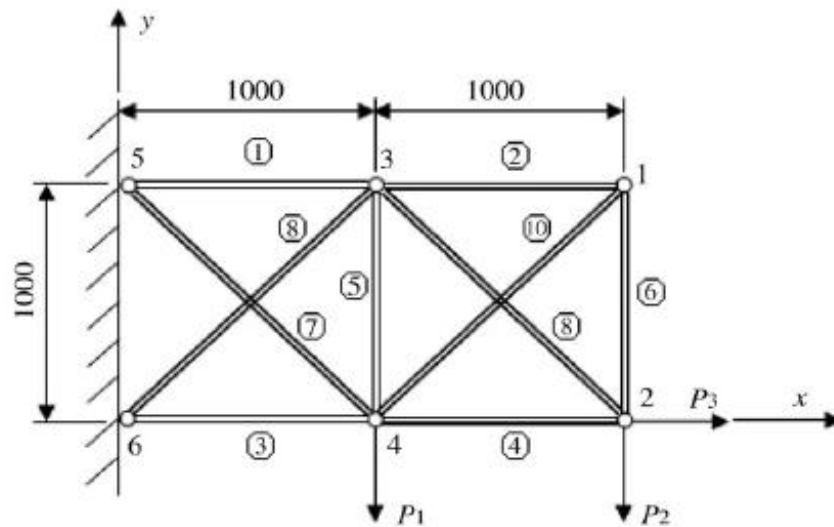


Figura 1.1: Modelo estructural de una cercha de diez barras

las condiciones necesarias para poder verificar la existencia de la solución es más simple realizarlo a través de un método computacional. En la próxima sección se exponen los conceptos necesarios para poder comprender como funcionan estos métodos computacionales.

1.3. Métodos computacionales

En esta sección se diseñan dos algoritmos para el tratamiento de los problemas de optimización multi-objetivo intervalo-valuado con restricciones intervalo valuadas. El diseño está soportado en la recopilación de los resultados obtenidos en los últimos cinco años sobre algoritmos evolutivos, [34], [21], [23]. En particular, el tema de las restricciones del problema de optimización intervalo-valuado, las cuales están descritas por funciones intervalo valuadas, es una adaptación del trabajo hecho por Coello en [5].

Antes de dar cuenta de los algoritmos mencionados, se expondrá una recopilación de algunos resultados sobre algoritmos evolutivos.

Los paradigmas de computación natural se han convertido en una buena opción al momento de resolver problemas de optimización, donde los algoritmos clásicos no han dado buenos resultados, debido a la complejidad y estructura de los problemas. Estos algoritmos de computación natural se pueden clasificar en tres categorías: Epigénesis, Filogenia y Ontogenia [23].

El grupo de Epigénesis son algoritmos que requieren aprender para resolver problemas complejos, en esta categoría se encuentran las Redes Neuronales Artificiales, las cuales simulan el cerebro humano. En el grupo de la Filogenia se encuentran los algoritmos en los cuales los individuos deben competir para sobrevivir de acuerdo con su capacidad de ajuste, estos son los llamados Algoritmos Evolutivos. Por último, encontramos el grupo Ontogenia, donde los individuos se adaptan con base a lo que sucede en el entorno, los PSO (Particle Swarm Optimization) pertenecen a esta categoría [23].

Para este trabajo se seleccionaron dos tipos de Heurísticos, Algoritmos Evolutivos, exactamente SPEA2 (Strength Pareto Evolutionary Algorithm 2), y PSO, que han comprobado ser efectivos y eficientes en solucionar diferentes tipos de problemas de optimización, aunque según Wolpert y Macready, en "No Free Lunch Theorems for Optimization"(NFL) [29], no son técnicas universales para solucionar cualquier tipo de problema de optimización. En las siguientes secciones se explicará en que consiste cada uno de ellos, además de presentarlos en una notación matemática, la cual permite hacer un estudio más formal de los mismos.

1.3.1. Algoritmos Evolutivos

Encontrar el conjunto de Pareto puede ser computacionalmente costoso y frecuentemente infactible, por la complejidad de los métodos exactos para encontrarlo. Por esta razón, numerosas estrategias de búsqueda estocástica, tales como Algoritmos Evolutivos, Búsqueda Tabú, Recocido Simulado y Colonia de Hormigas han sido desarrollados: estos usualmente no garantizan encontrar los valores óptimos, pero si proporcionar buenas soluciones que se encuentran muy cercanas a las óptima. [34]. Los Algoritmos Evolutivos se han presentado como una técnica robusta para resolver problemas de optimización Multi-Objetivo porque generan soluciones cercanas a la frontera de Pareto de forma efectiva y eficiente. Además, tienen ventajas sobre otras técnicas porque no es necesario modificar el dominio del problema para resolverlo, el proceso de búsqueda es fácil de desarrollar, entender y probar de una forma natural para una aplicación dada [5].

1.3.1.1. Conceptos Básicos de Algoritmos Evolutivos

En esta sección se presentan algunos términos usualmente utilizados en los Algoritmos Evolutivos. Análogamente a la evolución natural las soluciones candidatas son llamadas *individuos* y el conjunto de soluciones candidatas es llamado *población*, cada individuo representa una posible solución. Los individuos son usualmente representados por cadenas de caracteres que corresponden al genotipo biológico, el cual está compuesto por uno o más *cromosomas*, a su vez compuestos de *genes*, los cuales toman ciertos valores, *alelos*, de algún alfabeto genético. También, como en la naturaleza sobre la población, se aplican Operadores de Evolución para poder generar soluciones más adaptadas, es decir con un ajuste mayor. Los tres operadores usualmente utilizados son *mutación*, *apareamiento* y *selección* [34] [5].

En la selección, se distingue entre la de apareamiento y la de *ambiente*. La primera determina cuales soluciones candidatas serán tomadas para la variación, lo cual se hace usualmente con un método aleatorio. La segunda selección, se realiza para determinar cuales soluciones almacenadas continuarán.

El operador de mutación, toma soluciones candidatas, ya sea sistemáticamente o aleatoriamente, para ser modificadas y así generar soluciones potencialmente mejores [34].

El operador de apareamiento o reproducción usualmente consiste en dos pasos: asignación del ajuste y elección. En el primer paso, cada individuo de la población actual es evaluado en el espacio objetivo y se le asigna un valor escalar, el *ajuste*, que refleja su capacidad de adaptación o calidad. Luego, se eligen aleatoriamente los individuos que deben reproducirse de acuerdo con su valor de ajuste, esto se realiza usualmente con el método del torneo de selección[34].

Cada iteración es llamada *generación*, y frecuentemente se define un número máximo de iteraciones como criterio de parada; pero también puede ser que la población ya no varíe o se de la existencia de un individuo lo suficientemente calificado. Al final, el algoritmo proporciona los mejores individuos, los cuales son cercanos a la frontera de Pareto[34].

Encontrar la aproximación al conjunto de Pareto es por si mismo multi-objetivo, porque se debe minimizar la distancia de las soluciones generadas al conjunto de Pareto y maximizar la diversidad del conjunto no-dominado. Los algoritmos evolutivos multi-objetivo (Multi-Objective Evolutionary Algorithm - MOEA -) tienen como meta los dos objetivos anteriores a través de los siguientes aspectos: asignación del ajuste, preservación de la diversidad y elitismo, este último para prevenir que soluciones no-dominadas se pierdan durante el proceso[34].

A continuación se presenta las definiciones formales para estos conceptos, dadas por Merkle y Lamont en [15].

Definición 1.3.1. (Función Decodificadora): Sea I un conjunto no vacío y $f : \mathbb{R}^n \rightarrow \mathbb{R}$ (la función objetivo). Si $g : I \rightarrow \mathbb{R}^n$ es total, g es llamada función decodificadora

Definición 1.3.2. (Función de Ajuste): Sea I un conjunto no vacío, $g : I \rightarrow \mathbb{R}^n$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $h : \mathbb{R} \rightarrow \mathbb{R}$. Entonces $\Phi = h \circ f \circ g$ es llamada función de ajuste. Donde D es la función de decodificación, f la función a optimizar y h la función de ajuste escalada.

Definición 1.3.3. (Transformación de la Población): Sea I un conjunto no vacío, $\mu, \mu' \in \mathbb{Z}^+$ (el tamaño de la población padre y descendencia respectivamente). Una transformación $T : I^\mu \rightarrow I^{\mu'}$ es llamada una transformación de la población. Si $T(P) = P'$, P es llamado población padre y P' población descendiente. Si $\mu = \mu'$, entonces son llamados el tamaño de la población simplemente.

Definición 1.3.4. (Transformación Aleatoria de la Población): Sea I un conjunto no vacío, $\mu \in \mathbb{Z}^+$, Ω un conjunto (el espacio muestral). Una función $R : \Omega \rightarrow T(I^\mu, \cup_{\mu' \in \mathbb{Z}^+} I^{\mu'})$ es llamada una transformación aleatoria de la población.

Definición 1.3.5. (Operador Evolutivo): Sea I un conjunto no vacío, $\mu \in \mathbb{Z}^+$, \mathbb{X} un conjunto (el espacio de parámetros), Ω un conjunto. Una asignación

$$X : \mathbb{X} \rightarrow T(\Omega, T(I^\mu, \cup_{\mu' \in \mathbb{Z}^+} I^{\mu'})) \quad (1.12)$$

es llamado operador evolutivo. El conjunto de operadores evolutivos de la forma 1.12 es denotado $EVOP(I, \mu, \mathbb{X}, \Omega)$.

Definición 1.3.6. (Operador de Cruce): Sea $r \in EVOP(I, \mu, \mathbb{X}, \Omega)$. Si existe $P \in I^\mu$, $\Theta \in \mathbb{X}$ y $\omega \in \Omega$, tal que al menos un individuo en la población descendiente $r_\Theta(P)$ depende de más de un individuo de P entonces r es llamado operador de cruce.

Definición 1.3.7. (Operador de Mutación): Sea $m \in EVOP(I, \mu, \mathbb{X}, \Omega)$, Si para todo $P \in I^\mu$, todo $\Theta \in \mathbb{X}$ y todo $\omega \in \Omega$, cada individuo en la población descendiente $m_\Theta(P)$ depende a lo sumo de un individuo de P entonces m es llamado operador de mutación.

Definición 1.3.8. (Operador de Selección): Sea $s \in EVOP(I, \mu, \mathbb{X} \times T(I, \mathbb{R}), \Omega)$, Si para todo $P \in I^\mu$, todo $\Theta \in \mathbb{X}$ y toda función de ajuste $\Phi : I \rightarrow \mathbb{R}$, s satisface $a \in S_{(\Theta, \Phi)}(P) \Rightarrow a \in P$, entonces s es llamado operador de selección.

Definición 1.3.9. (Algoritmo Evolutivo): Sea I un conjunto no vacío (espacio de individuos), $\{\mu^{(i)}\}_{i \in \mathbb{N}}$ una secuencia en \mathbb{Z}^+ (los tamaños de las poblaciones padres), $\{\mu'^{(i)}\}_{i \in \mathbb{N}}$ una secuencia en \mathbb{Z}^+ (los tamaños de las poblaciones descendientes), $\Phi : I \rightarrow \mathbb{R}$ una función de ajuste, $\iota : \cup_{i=1}^{\infty} (I^\mu)^i \rightarrow \{\text{true}, \text{false}\}$ (el criterio de parada), $\chi \in \{\text{true}, \text{false}\}$, r una secuencia $\{r^{(i)}\}$ de operadores de cruce $r^{(i)} : \mathbf{X}_r^{(i)} \rightarrow T(\Omega_r^{(i)}, T(I^{\mu^{(i)}}), I^{\mu'^{(i)}})$, m una secuencia $\{m^{(i)}\}$ de operadores de mutación $m^{(i)} : \mathbf{X}_m^{(i)} \rightarrow T(\Omega_m^{(i)}, T(I^{\mu^{(i)}}), I^{\mu'^{(i)}})$, s una secuencia $\{s^{(i)}\}$ de operadores de selección $s^{(i)} : \mathbf{X}_s^{(i)} \times T(I, \mathbb{R}) \rightarrow T(\Omega_s^{(i)}, T((I^{\mu^{(i)} + \chi \mu'^{(i)}}), I^{\mu'^{(i)}}))$, $\Theta_r^{(i)} \in \mathbf{X}_r^{(i)}$ (los parámetros de cruce), $\Theta_m^{(i)} \in \mathbf{X}_m^{(i)}$ (los parámetros de mutación), $\theta_s^{(i)} \in \mathbf{X}_s^{(i)}$ (los parámetros de selección). El algoritmo presentado a continuación es llamado *Algoritmo Evolutivo*

Algorithm 1.3.1: ALGORITMOEVOLUTIVO()

```

t ← 0
P(0) ← {a1(0), ..., aμ(0)} ∈ Iμ
while ι(P(0), ..., P(t)) ≠ true
    comment: Cruce
    P'(t) ← r(t)Θr(t)(P(t))
    comment: Mutacion
    P''(t) ← m(t)Θm(t)(P'(t))
    comment: Seleccion
do {
    P'(t) ← r(t)Θr(t)(P(t))
    if χ
    then P(t+1) ← s(t)(θs(t), Φ)(P''(t))
    else P(t+1) ← s(t)(θs(t), Φ)(P''(t) ∪ P(t))
    t ← t + 1

```

1.3.1.2. Clasificación de Técnicas de Búsqueda y Decisión

Existen diferentes formas de clasificar los algoritmos evolutivos multi-objetivo, en este trabajo se utilizará la clasificación con base en la técnica utilizada por el algoritmo al momento de realizar la búsqueda y manejar los múltiples objetivos [5]:

1.3.1.2.1. Técnicas A Priori Requieren que el *Decisor* determine la importancia relativa de cada uno de los objetivos. Usualmente se utilizan estrategias de agregación o lexicográficas [28, 5].

La estrategia lexicográfica, con base al orden definido por el *Decisor*, optimiza los objetivos en forma secuencial. También se puede realizar en forma aleatoria; está técnica optimiza un objetivo a la vez.

La estrategia de agregación consiste en convertir los objetivos en una sola función objetivo parametrizada, la cual puede ser lineal o no lineal. A continuación se presentan las agregaciones más comunes:

Weighted-sum: Combinación lineal de los objetivos, el Decisor asigna pesos positivos a cada uno de los objetivos para determinar la importancia relativa de cada objetivo. Se representa matemáticamente de la siguiente forma:

$$\min \sum_{i=1}^k w_i f_i(x) \quad (1.13)$$

donde $w_i \geq 0$ son los pesos determinados por el Decisor para el objetivo i . Además, se suelen normalizar los pesos así:

$$\sum_{i=1}^k w_i = 1 \quad (1.14)$$

Multiplicativa: Las funciones objetivos son combinadas a través de una productoria. La forma más general es:

$$\min \prod_{i=1}^k f_i(x) \quad (1.15)$$

Vector Objetivo: El Decisor establece la meta esperada para cada uno de los objetivos y la distancia de cada función a la meta se convierte en la métrica de ajuste. Es representada así:

$$\min [d(f_i(x), g)] W^{-1} \quad (1.16)$$

donde $g = (g_1, \dots, g_k)$ es un vector que representa las metas esperadas y W es una matriz de pesos.

1.3.1.2.2. Técnicas Progresivas Estas técnicas son interactivas, el Decisor debe interactuar durante la ejecución del programa, en cada ciclo se decide y se busca o viceversa, aplicando técnicas *a priori* o *a posteriori* [28, 5]

1.3.1.2.3. Técnicas A Posteriori Las técnicas *a posteriori* buscan la frontera de Pareto, el Decisor participa después de que se realiza la búsqueda. A continuación se presentan las técnicas más usuales.

Muestreo Independiente: En cada iteración se optimiza una función de agregación diferente, donde los pesos asignados a los objetivos en cada iteración son diferentes. Se diferencia de la agregación *a priori* en que los pesos no son asignados con anterioridad.

Selección por Criterios: La población es dividida en fracciones, donde en cada fracción se evalúa un objetivo diferente para seleccionar los individuos.

Selección Basada en Pareto: La idea de los basados en Pareto es calcular la dominancia de cada individuo. Algunos enfoque utilizan el número de individuos que dominan a cada individuo para determinar el valor del ajuste. Otro usan la profundidad de la dominancia, donde la población es dividida en diferente frentes y la profundidad se determina por el frente al cual pertenece el individuo. También es utilizado el contador de dominancia, es decir el número de individuos dominados. Esta última será la técnica que se usará en este trabajo.

1.3.1.3. Técnicas para Preservación de la Diversidad

Como se mencionó anteriormente un objetivo de los MOEA es conservar la diversidad de la población para encontrar una aproximación de la frontera de Pareto. Para lograr este objetivo se han desarrollado varias técnicas, las cuales se mencionan a continuación [34, 5].

Enfoque del Vector de Pesos: Se conserva un conjunto de vectores en el espacio objetivo para mantener diversidad de puntos de la frontera de Pareto. Esto se logra cambiando los pesos de los objetivos en el proceso de búsqueda, para que diferentes direcciones se definan en el momento en que los individuos se mueven en su vecindad.

Ajuste Compartido/ Enfoque de Nichos: Se define el tamaño de un vecindario o nicho, el tamaño de este es determinado por el radio del nicho σ_{share} , y se controla el número de individuos que se encuentran en un vecindario. Para determinar la densidad del vecindario existen diferentes enfoques, a seguir:

- *Kernel:* Definen el vecindario de un individuo en términos de la también llamada función Kernel K , la cual toma la distancia otro punto como un argumento. En la práctica, para cada individuo la distancia d_i a otro individuo i es calculada y después se aplica K resultando $K(d_i)$. La suma de los valores de la función K representa la densidad estimada para cada individuo.
- *Vecino más cercano:* Toman la distancia de un punto dado a su k -ésimo vecino más cercano con el objetivo de estimar la densidad en su vecindario. Usualmente, el estimador es una función de la inversa de la distancia. Esta técnica será la que se utilice en el algoritmo que se propone, porque con ella se conserva mayor diversidad, debido a que se tendrá mayor cobertura de la frontera de Pareto, y así no se aglomeran los individuos alrededor de un punto.
- *Histogramas:* La densidad alrededor de un individuo es estimada por el número de individuos en la misma hipercaja o hipermalla.

Clustering: Se seleccionan las soluciones con base en una métrica de aglomeración medida en el espacio objetivo.

1.3.1.4. Elitismo

El elitismo trata de evitar la pérdida de buenas soluciones durante el proceso de optimización. Para solucionar este problema, por un lado se tiene la población y la descendencia a la cual se le aplica un procedimiento de selección determinístico. Por otro lado, una segunda población, también llamada *archive*, en la cual se encuentran las soluciones más prometedoras de cada generación, las soluciones no-dominadas [34, 5].

Como el recurso de memoria es usualmente restringido se deben establecer criterios para determinar cuáles soluciones deben mantenerse en él. El criterio de dominancia es el más utilizado; pero este usualmente no es suficiente porque el conjunto de Pareto en problemas continuos contiene infinitas soluciones, por tal motivo otros criterios son introducidos, como la información de densidad o el tiempo transcurrido en el archivo para tomar esta decisión [34].

La mayoría de los MOEA combinan los criterios de densidad y dominancia para determinar cuáles individuos deben estar en el archivo, conjunto no-dominado, en cada generación [34].

1.3.1.5. Representación del Cromosoma

Las representaciones más comunes para el cromosoma en los algoritmos evolutivos son: forma binaria, números reales o punto flotante y vectores de permutación; realmente la representación del cromosoma debe depender del tipo de problema, en [29] se muestra como un algoritmo que es muy bueno para un tipo de problema es muy malo para otro.

A continuación se presentan los operadores de cruce y mutación más comunes para la representación en números reales.

1.3.1.5.1. Operadores de Cruce

- **Cruce Simple**

Este operador es muy similar al operador de cruce simple binario, donde dos individuos son seleccionados y sus descendientes son la combinación convexa de ellos, después de una posición definida; es decir para dos padres $v = (v_1, \dots, v_n)$ y $w = (w_1, \dots, w_n)$, si son cruzados en la posición i -ésima, los descendientes generados son $v' = (v_1, \dots, v_i, \lambda w_{i+1} + (1 - \lambda)v_{i+1}, \dots, \lambda w_n + (1 - \lambda)v_n)$ y $w' = (w_1, \dots, w_i, \lambda v_{i+1} + (1 - \lambda)w_{i+1}, \dots, \lambda v_n + (1 - \lambda)w_n)$ donde $\lambda \in [0, 1]$.

- **Cruce Aritmético Simple**

Para dos padres $v = (v_1, \dots, v_n)$ y $w = (w_1, \dots, w_n)$, si son cruzados en la posición i -ésima, la descendencia resultante es

$$v' = (v_1, \dots, v'_i, \dots, v_n)$$

$$w' = (w_1, \dots, w'_i, \dots, w_n)$$

donde $v'_i = \lambda w_i + (1 - \lambda)v_i$, $w'_i = \lambda v_i + (1 - \lambda)w_i$ y λ es un número aleatorio escogido así:

$$\lambda = \begin{cases} [\text{máx}(\alpha, \beta), \text{mín}(\gamma, \delta)] & \text{si } v_i > w_i \\ [0, 0] & \text{si } v_i = w_i \\ [\text{máx}(\gamma, \delta), \text{mín}(\alpha, \beta)] & \text{si } v_i < w_i \end{cases} \quad (1.17)$$

donde

$$\alpha = (l(w_i) - w_i)/(v_i - w_i) \quad (1.18)$$

$$\beta = (u(v_i) - v_i)/(w_i - v_i) \quad (1.19)$$

$$\gamma = (l(v_i) - v_i)/(w_i - v_i) \quad (1.20)$$

$$\delta = (u(w_i) - w_i)/(v_i - w_i) \quad (1.21)$$

Aquí $l(w_i)$ y $u(w_i)$ denotan el límite inferior y superior de w_i respectivamente.

- **Cruce Aritmético Completo**

Para dos padres v y w , el operador genera dos descendientes v' y w' , los cuales son una combinación convexa de sus padres, así:

$$v' = \lambda w + (1 - \lambda)v \text{ y } w' = \lambda v + (1 - \lambda)w, \lambda \in [0, 1] \quad (1.22)$$

El operador utiliza un número aleatorio $\lambda \in [0, 1]$, cuando $\lambda = \frac{1}{2}$, el operador es llamado operador de cruce promedio.

- **Cruce Heurístico**

Utiliza los valores de la función objetivo para determinar la dirección de búsqueda y generar un único descendiente z de dos padres v y w , así:

$$z = r(w - v) + w \quad (1.23)$$

donde r es un número aleatorio entre 0 y 1 y el $f(w) < f(v)$ en un problema de minimización.

1.3.1.5.2. Operadores de Mutación

- **Mutación Uniforme**

El operador selecciona aleatoriamente un componente $i \in 1, \dots, n$ de el vector $v = (v_1, \dots, v_i, \dots, v_n)$ y genera $v' = (v_1, \dots, v'_i, \dots, v_n)$. Aquí v'_i es un número aleatorio entre $l(v_i)$ y $u(v_i)$.

- **Mutación Límite**

Este operador es una variación de la mutación uniforme, donde v'_i es el valor $u(v_i)$ o $l(v_i)$, cada uno con igual probabilidad.

- **Mutación No Uniforme**

Este operador es diseñado para afinar los resultados y lograr una mayor precisión. Para un padre dado v , si el elemento i es seleccionado para mutación, el descendiente resultante es $v'_i = (v_1, \dots, v'_i, \dots, v_n)$, donde:

$$v'_i = \begin{cases} v_i + \Delta(t, u(v_i) - v_i) & \text{si } r < 0,5 \\ v_i - \Delta(t, v_i - u(v_i)) & \text{si } r \geq 0,5 \end{cases} \quad (1.24)$$

La función $\Delta(t, y)$ retorna un valor en el rango $[0, y]$ tal que la probabilidad de $\Delta(t, y)$ tiende a 0 a medida que avanzan las generaciones. La función $\Delta(t, y)$ se define como:

$$\Delta(t, y) = yr(1 - \frac{t}{T})^b \quad (1.25)$$

Donde r es un número aleatorio entre 0 y 1, T es el número máximo de generaciones, y b es un parámetro que determina el grado de no uniformidad.

Cuando el operador es aplicado a todo el vector se comporta como Mutación Uniforme Completa.

1.3.1.6. Técnicas para el Manejo de Restricciones

Existen muchas técnicas en los Algoritmos Evolutivos que se han desarrollado para resolver problemas de optimización multi-objetivo; sin embargo, los métodos no indican como tener presente las restricciones del problema de optimización, de tal forma que se logre un mayor número de soluciones candidatas que sean factibles. Por tal motivo, se han desarrollado técnicas que incorporan las restricciones dentro del proceso de búsqueda y decisión, disminuyendo la posibilidad de que las soluciones finales sean no factibles. Estas técnicas se clasifican en [4]:

1.3.1.6.1. Funciones de Penalización: La penalización es la técnica más utilizada, consiste en transformar el problema en uno de optimización sin restricciones. Para lograrlo, se suma o resta, a la función objetivo, cierta cantidad de acuerdo con la cantidad de restricciones violadas.

Los Algoritmos Evolutivos usualmente utilizan una función de penalización de la forma:

$$\Phi(x) = f(x) \pm \left[\sum_{i=1}^n r_i \times G_i + \sum_{i=1}^p c_i \times L_i \right] \quad (1.26)$$

donde $\Phi(x)$ es la nueva función objetivo a ser optimizada, G_i y H_i son funciones de las restricciones g_i y h_i respectivamente y r_i y c_i son constantes positivas llamadas *factores de penalización*.

La forma más común de las funciones G_i y H_i es:

$$G_i = \max\{0, g_i(x)\}^\beta \quad (1.27)$$

$$H_i = \max\{0, h_i(x)\}^\gamma \quad (1.28)$$

donde normalmente β y γ son 1 ó 2.

Existen varios tipos de funciones de penalización, los más utilizados son los siguientes:

Penalización de Muerte

Es probablemente la forma más fácil de manejar las restricciones, porque las soluciones que violan las restricciones son rechazadas y se generan de nuevo. Por lo tanto, no es necesario calcular el grado de infactibilidad de estas soluciones. Sin embargo, este tipo de penalización no es aconsejable porque puede tener problemas en regiones factibles muy pequeñas y se pierde información de los puntos infactibles, los cuales podrían conducir a mejores soluciones.

Penalización Estática

Los factores de penalización son independientes de la generación, es decir, permanecen constantes durante todo el proceso.

Un ejemplo de este tipo de penalización es el propuesto por Homaifar, Lai y Qi [8] en el cual se define el nivel de violación de las restricciones y los factores de penalidad asociados.

$$fitness(x) = f(x) + \sum_{i=1}^m (R_{k,i} \times \max[0, g_i(x)]^2) \quad (1.29)$$

donde $R_{k,i}$ son los coeficientes de penalización utilizados, m es el número total de restricciones, $f(x)$ es la función objetivo no penalizada, y $k = 1, 2, \dots, l$, donde l es número de niveles de violación definidos por el usuario.

Las principales desventajas de esta penalización son: los factores de penalización son dependientes del problemas y son estáticos durante todo el proceso de búsqueda.

Penalización Dinámica

Los factores de penalización dependen de la generación, usualmente se incrementan en cada generación, para ser más estrictos, al momento de seleccionar las soluciones candidatas.

Un ejemplo es el desarrollado por Joines y Houck [12] donde la función de ajuste penalizada para los individuos de la generación t es:

$$fitness(x) = f(x) + (C \times t)^\alpha \times SVC(\beta, x) \quad (1.30)$$

donde C , α y β son constantes definidas por el usuario, los autores recomiendan utilizar $C = 0,5$, $\alpha = 1, 2$ y $\beta = 1, 2$. $SVC(\beta, x)$ esta definida como:

$$SVC(\beta, x) = \sum_{i=1}^n D_i^\beta(x) + \sum_{j=1}^p D_j(x) \quad (1.31)$$

$$D_i(x) = \begin{cases} 0 & g_i(x) \leq 0 \\ |g_i(x)| & g_i(x) > 0 \end{cases} \quad (1.32)$$

$$D_j(x) = \begin{cases} 0 & -\epsilon \leq h_j(x) \leq \epsilon \\ |h_j(x)| & |h_j(x)| > \epsilon \end{cases} \quad (1.33)$$

En la función dinámica la penalidad se va incrementando a medida que aumentan las generaciones.

Este tipo de penalización trabaja mucho mejor que la estática, porque no hay que definir factores de penalización.

Penalización Adaptativa:

Este método fue desarrollado por Bean y Hadj-Aloune [6], el cual se retroalimenta del proceso de búsqueda para aumentar o disminuir la penalización. El ajuste de cada individuo es evaluado por la formula:

$$fitness(x) = f(x) + \lambda(t) \left(\sum_{i=1}^n g_i^2(x) + \sum_{i=1}^p |h_i(x)| \right) \quad (1.34)$$

donde $\lambda(t)$ es actualizado cada generación t así:

$$\lambda(t+1) = \begin{cases} (\lambda(t)/\beta_1) & \text{caso 1} \\ \beta_2 * \lambda(t) & \text{caso 2} \\ \lambda(t) & \text{otro caso} \end{cases} \quad (1.35)$$

Los casos 1 y 2 denotan la situación del mejor individuo en las últimas k generaciones, el caso 1 es que siempre estuvo en la región factible, y el caso 2 que nunca estuvo, $\beta_1 > \beta_2 > 1$ y $\beta_1 \neq \beta_2$. Esto quiere decir, que la función de penalización disminuye si los mejores individuos de las últimas k generaciones han estado en la región factible e aumenta sino han sido soluciones factibles. Si algunos han sido factibles y otros no, la función de penalización no cambia.

La mayor dificultad de ésta es determinar el valor de los parámetros para que sean apropiados.

1.3.1.6.2. Separación de las Restricciones y Objetivos Al contrario de la técnica de funciones de penalización, las cuales combinan el valor de la función objetivo y de las restricciones para determinar el ajuste, esta técnica trata por separado las restricciones y la función objetivo. Algunos ejemplos de ellos son:

Co-evolución: Utiliza dos poblaciones, restricciones y soluciones, que interactúan una con la otra de una forma similar al modelo depredador-presa [17].

Superioridad de los puntos factibles: Asigna valores de ajuste más altos a las soluciones factibles [19].

Memoria de Comportamiento: Trata de satisfacer, secuencialmente, las restricciones del problema [4].

Uso de los conceptos de optimización multi-objetivo: Redefine el problema de optimización, adicionando las restricciones como objetivos, así el problema se convierte en multi-objetivo con m objetivos más, donde m es el número de restricciones. A continuación se presentan los tipos más populares en los últimos años:

- **Optimización bajo Restricciones por Algoritmos Genéticos de Optimización Multi-objetivo:** Los individuos son clasificados con base en la suma de restricciones violadas teniendo en cuenta el concepto de dominancia de Pareto [25].
- **Clasificación basada en Fuerza de Pareto:** Se clasifican los individuos con base en los individuos dominados y la suma de las restricciones violadas, se seleccionan para la siguiente generación los individuos con menor Fuerza de Pareto y menor suma de restricciones [7].

Un algoritmo fundamental en este trabajo es el siguiente:

1.3.2. Strength Pareto Evolutionary Algorithm - SPEA2

El Algoritmo Evolutivo de Fuerza de Pareto (SPEA - Strength Pareto Evolutionary Algorithm) fue introducido por Eckart Zitzler y Lothar Thiele como una técnica para encontrar aproximaciones de la frontera de Pareto y ha sido evaluada favorablemente en comparación con otros MOEA. El SPEA se diseñó con el objetivo de eliminar los individuos potencialmente débiles de la generación predecesora e incorporar los más adaptados, de tal manera que se tenga una población *fuerte*, lo cual se logra aplicando el elitismo a través de un conjunto de soluciones no-dominadas, en el cual se copian los individuos no-dominados de cada generación, donde se seleccionan con base en Pareto. Después Eckart Zitzler y Lothar Thiele lo mejoraron y lo denominaron SPEA2 [34, 33, 5], a continuación se introducen las variaciones que lo hace más potente que el SPEA.

- Mejora la asignación del ajuste, porque toma no sólo la cantidad de individuos que domina sino también la cantidad de individuos que lo dominan.
- Utiliza la técnica de densidad de vecino más próximo, lo cual guía de una manera más precisa el proceso de búsqueda.
- Tiene un método de truncamiento de la población que garantiza la preservación de las soluciones límites.

El algoritmo propuesto por Zitzler y Lothar Thiele se presenta a continuación [33]

Algorithm 1.3.2: SPEA2(N : Tamanno de la Poblacion, T : Numero de Generaciones)

```

Generar poblacion inicial  $P_0$ 
 $A_0 \leftarrow \Phi$ 
for  $t \leftarrow 0$  to  $T$ 
    {
    Calcular  $F(i) \forall i \in P_t \cup A_t$ 
     $A_{t+1} \leftarrow \{i | i \in P_t \cup A_t \wedge F(i) < 1\}$ 
    if  $|A_{t+1}| < N$ 
        then Completar la poblacion con los  $|A_{t+1}| - N$  mejores individuos dominados de  $P_t \cup A_t$ 
    do {
        else Aplicar operador de truncamiento con base al estimador de densidad
        Aplicar operador de seleccion para apareamiento
         $P_{t+1} \leftarrow r_{\Theta_r^{(i)}}^{(t)}(P_t)$ 
         $P_{t+1} \leftarrow m_{\Theta_m^{(i)}}^{(t)}(P_{t+1})$ 
    }
return  $(A)_t$ 

```

A continuación se presentarán cada una de las técnicas utilizadas en la función de ajuste, preservación de la diversidad, búsqueda y selección.

1.3.2.1. Función de Ajuste y Preserveración de la Diversidad

La función de ajuste se hace con base en la dominancia de Pareto y el estimador de densidad de aglomeración. Primero se calcula para cada individuo i la fuerza asignada $S(i)$, la cual se representa con base en la cantidad de soluciones que domina:

$$S(i) = |j | j \in P_t \cup A_t \wedge i > j| \quad (1.36)$$

donde $| \cdot |$ denota la cardinalidad del conjunto y el símbolo $>$ corresponde a la relación de dominancia de Pareto. Con base en los valores de S , se calcula el ajuste *crudo* de cada individuo i de la siguiente forma:

$$R(i) = \sum_{j \in P_t \cup A_t, j > i} S(j) \quad (1.37)$$

Este ajuste crudo es determinado por las fuerzas de los dominantes tanto de la población como en el conjunto no-dominado. Es importante notar que los individuos no dominados tienen un $R(i) = 0$, y los individuos que son dominados por más individuos tienen el $R(i)$ más alto.

La asignación del ajuste crudo provee un ordenamiento basado en el mecanismo de nichos sobre la dominancia de Pareto, que puede fallar si muchos individuos no se dominan los unos a los otros. Por tal motivo, el estimador densidad es incorporado en el ajuste como forma de diferenciar los individuos con valores de ajuste crudo igual y de preservar la diversidad de las soluciones. La técnica de estimación de densidad utilizada en SPEA2 es una adaptación del método del k-ésimo vecino más cercado, donde la densidad de un punto es una función de la distancia a el k-ésimo punto más cercano. Luego, se toma el inverso de la distancia al k-ésimo vecino más cercano como la estimación de la densidad. Para lograr esto, se calcula y almacena la distancia de cada individuo i a todos los individuos j del

conjunto no dominado y de la población. Después, se ordena la lista en forma ascendente y el k-ésimo elemento proporciona la distancia buscada, la cual se denota como σ_i^k . Una práctica muy común es utilizar la raíz cuadrada del tamaño de la muestra para k ; sin embargo, $k = 1$ es suficiente para una implementación eficiente. Ahora, la densidad $D(i)$ correspondiente al individuo i se define como:

$$D(i) = \frac{1}{\sigma_i^k + 2} \quad (1.38)$$

En el denominador, se adiciona dos para asegurar que el valor sea mayor que cero y que $D(i) < 1$. Finalmente, se adiciona $D(i)$ al valor del ajuste crudo $R(i)$ de cada individuo i , proporcionando su ajuste $F(i)$:

$$F(i) = R(i) + D(i) \quad (1.39)$$

1.3.2.2. Selección Siguiente Generación

Durante el proceso de selección de la siguiente generación, el primer paso es copiar todos los individuos no-dominados, es decir, aquellos que tienen un ajuste menor que uno, del conjunto no dominado y la población a el conjunto no dominado de la siguiente generación.

$$A_{t+1} = \{i | i \in P_t \cup A_t \wedge F(i) < 1\} \quad (1.40)$$

Si el frente no-dominado tiene una cantidad de individuos exactamente igual a la cantidad de individuos que deben estar en el conjunto no dominado ($|A_{t+1}| = N$), este paso es completado. De otro modo, puede darse una de dos situaciones: Que sea menor ($|A_{t+1}| < N$) o que sea mayor ($|A_{t+1}| > N$). Para el primer caso, los mejores $N - |A_{t+1}|$ individuos dominados de la población y el anterior conjunto no dominado son copiados al nuevo conjunto no dominado. Esto puede ser implementado ordenando el conjunto $P_t \cup A_t$, de acuerdo con el ajuste y copiar los primeros $N - |A_{t+1}|$ individuos i con $F(i) \geq 1$ a A_{t+1} . En el segundo caso, cuando el tamaño de los nuevos individuos no-dominados excede N , se trunca el conjunto removiendo individuos hasta que se llega a N . Así, en cada iteración se remueve el individuo i , tal que $i \leq_d j$ para todos los $j \in A_{t+1}$ con

$$i \leq_d j : \Leftrightarrow \forall 0 < k < |A_{t+1}| : \sigma_i^k = \sigma_j^k \vee \exists 0 < k < |A_{t+1}| : \left[(\forall 0 < l < k : \sigma_i^l = \sigma_j^l) \wedge \sigma_i^k < \sigma_j^k \right] \quad (1.41)$$

donde σ_i^k denota la distancia de i a el k-ésimo vecino mas cercano en A_{t+1} . En otras palabras, el individuo con las distancia menor a otros individuos es escogido para la siguiente generación; si hay varios individuos con las distancia menor el empate se evalúa por la segunda distancia más pequeña y así sucesivamente. La técnica de truncamiento se muestra en la figura 1.2.

Este es el otro algoritmo que es parte fundamental de este trabajo.

1.3.3. MultiObjective Particle Swarm Optimization - MOPSO

PSO es un algoritmo de búsqueda basado en la simulación del comportamiento social de las aves dentro de una bandada, este fue propuesto por James Kennedy and Russell Eberhart. En este algoritmo cada solución es representada como una partícula, cada partícula se va moviendo en la región de búsqueda y cada movimiento se basa en sus experiencias y las de toda la población, está búsqueda es

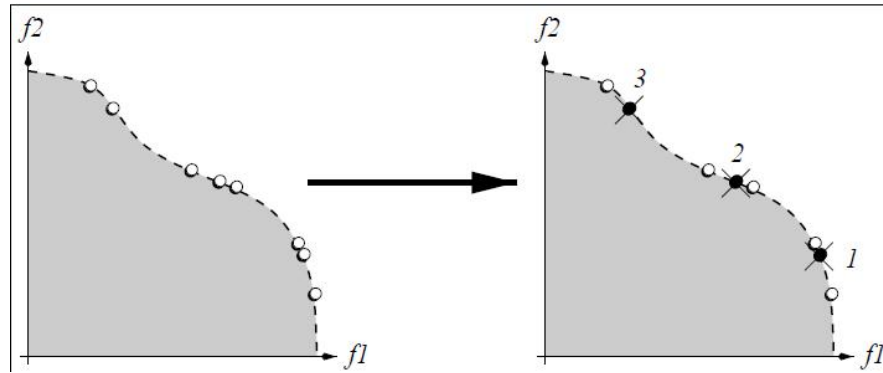


Figura 1.2: Ilustración del método de truncamiento utilizado en SPEA2. A la derecha, un conjunto no-dominado. A la izquierda, se representan las soluciones que se eliminan por el operador de truncamiento (suponiendo que $N = 5$).

con el objetivo de encontrar el óptimo global. Así la posición de cada partícula va cambiando con base en la velocidad, así:

$$X_{t+1}(i) = X_t(i) + V_t(i) \quad (1.42)$$

donde $X_t(i)$ representa la posición actual de la partícula i en el momento t y $V_t(i)$ la velocidad con la cual se desplazará en el espacio de búsqueda, esta velocidad esta definida por:

$$V_{t+1}(i) = V_t(i) + C_1 \times r_1 \times (X_{lb}(i) - X_t(i)) + C_2 \times r_2 \times (X_{gb} - X_t(i)) \quad (1.43)$$

donde r_1 y r_2 son número aleatorio entre $[0, 1]$, C_1 es el coeficiente de aceleración basado en la propia experiencia y C_2 es el coeficiente de aceleración basado en la experiencia social, $X_{lb}(i)$ representa la mejor posición en la cual ha estado ubicada la partícula i y X_{gb} la mejor posición en la cuál se han encontrado todas las partículas.

El algoritmo básico del PSO se presenta a continuación:

Algorithm 1.3.3: PSO()

```

for  $i \leftarrow 1$  to  $M - 1$ 
  (comment: Inicializa aleatoriamente la partícula  $i$ 
    $X_0(i) \leftarrow random$ 
   comment: Inicializa la velocidad de la partícula  $i$ 
    $V_0(i) \leftarrow 0$ 
  do (comment: Inicializa la mejor posición de la partícula  $i$ 
    $X_{lb}(i) \leftarrow X_0(i)$ 
   comment: Selecciona la mejor posición global
   if  $f(X_0(i)) < X_{gb}$ 
     then  $X_{gb} \leftarrow f(X_0(i))$ 
  while que no se cumpla el criterio de parada
    ( $V_{t+1}(i) \leftarrow V_t(i) + C_1 \times r_1 \times (X_{lb}(i) - X_t(i)) + C_2 \times r_2 \times (X_{gb} - X_t(i))$ 
     comment: Actualiza la posición de cada partícula
      $X_{t+1}(i) \leftarrow X_t(i) + V_t(i)$ 
     if  $f(X_t(i)) < f(X_{lb}(i))$ 
     do (comment: Evalua la función objetivo en cada partícula
       then  $X_{lb(i)} \leftarrow f(X_t(i))$ 
       if  $X_{lb}(i) < X_{gb}$ 
       comment: Actualiza el mejor global
       then  $X_{gb} \leftarrow X_{lb}(i)$ 

```

Para el caso multi-objetivo no hay una sola solución, es un conjunto de soluciones, es decir, la frontera de Pareto; al igual que los algoritmos evolutivos, el PSO para el caso multi-objetivo también tiene dos objetivos: maximizar la diversidad y minimizar la distancia a la frontera de Pareto. Por tal motivo se deben hacer algunos cambios al algoritmo básico en la selección de la *mejor* posición global y así poder satisfacer ambos objetivos durante el proceso de búsqueda. Como cada partícula no tiene una sola solución mejor sino un conjunto de soluciones, estas soluciones se deben almacenar en un archivo, en este se conservan las soluciones no-dominadas encontradas y es utilizado al momento de escoger la mejor solución.

Capítulo 2

Algoritmos Propuestos

En este trabajo se proponen dos algoritmos basados en el SPEA2 y el MOPSO para solucionar problemas multi-objetivo con restricciones bajo incertidumbre, esta incertidumbre es expresada a través de intervalos. Para poder solucionar este tipo de problemas hubo que realizar algunos cambios a los algoritmos y se reformuló el problema multi-objetivo adicionando un nuevo objetivo: minimizar el número de restricciones violadas, así el problema de optimización quedaría de la siguiente forma:

$$\begin{aligned} \min F^*(x) \\ x \in \Omega \end{aligned} \tag{2.1}$$

Donde $F^*(x) : \mathbb{R}^n \rightarrow I^{m+1}(\mathbb{R})$, es decir se adiciona un objetivo a la función original $F : \mathbb{R}^n \rightarrow I^m(\mathbb{R})$, este objetivo es con base en las restricciones y se define como $G^* : \mathbb{R}^n \rightarrow I(\mathbb{R}), x \rightarrow [G(x), G(x)]$, donde $G(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n, x \rightarrow |\{g_j(x) | (g_j(x) \not\leq 0), g_j : \mathbb{R}^n \rightarrow I(\mathbb{R}) \text{ es la restricción } j, \text{ con } j \in 1, \dots, p\}|$. Es decir, las restricciones fueron introducidas como un objetivo más a minimizar.

En la primera sección se presentará el SPEA2I - Strength Pareto Evolutionary Algorithm 2 for Interval-Valued Problems y en la siguientes sección el MOIPSO - Multi-Objective Interval-Valued Particle Swarm Optimization.

2.1. Strength Pareto Evolutionary Algorithm for Interval-valued Problems - SPEA2I

Las variaciones que se realizan al SPEA2 para adaptarlo a problemas multi-objetivos intervalo-valorados son incorporación del manejo de restricciones dentro del proceso selección de individuos a la siguiente generación e individuos no-dominados; además, la dominancia de Pareto y los operadores evolutivos son definidos con base en la teoría intervalo-valorada.

A continuación se presentará cada uno de los aspectos de los Algoritmos Evolutivos y la técnica utilizada en cada uno de ellos.

2.1.1. Representación del Cromosoma

La representación del cromosoma se hizo a través de números reales, porque son los más adecuados para los problemas multi-objetivo intervalo-valuados ya que permiten mayor precisión que la binaria y con tal representación es fácil diseñar los operadores evolutivos de cruce o apareamiento y mutación, es decir, $P \subseteq \mathbb{R}^n$. El operador de cruce utilizado es el aritmético simple con el cual se realiza una combinación lineal de cada par de individuos en cada elemento y el operador de mutación es el no uniforme con el cual se introduce mayor aleatoriedad en la búsqueda permitiendo mayor exploración de la región de búsqueda.

2.1.2. Función de Ajuste y Preserveración de la Diversidad

Para calcular la función de ajuste se realizan los mismos pasos que en el SPEA2: primero se calcula la fuerza de cada individuo con base a los individuos que dominan, el concepto de dominancia para $F * (x)$ es el mismo descrito en el conjunto $I^m(\mathbb{R})$, el cual también es utilizado al momento de determinar el ajuste crudo $R(i)$ de cada individuo, en la función de densidad al momento de calcular la distancia al k-ésimo vecino sólo se tienen en cuenta los objetivos originales y utilizando la distancia Hausdorff infinito, por último se calcula la función de ajuste $F(x) = R(i) + D(i)$.

2.1.3. Selección Siguiete Generación

En el proceso de selección, también se hicieron variaciones para tener en cuenta las restricciones, primero se incluyen los individuos no-dominados que no violan ninguna restricción.

$$A_{t+1} = \{i | i \in P_t \cup A_t \wedge F(i) < 1 \wedge |\{g_j(i) | (g_j(i) \not\leq 0)\}| = 0\} \quad (2.2)$$

Si el conjunto no-dominado tiene una cantidad de individuos menor ($|A_{t+1}| < N$), los mejores $N - |A_{t+1}|$ individuos dominados de la población que no violen restricciones, si no se ha completado la cantidad de la población, se completa con los individuos dominados que violan restricciones y se encuentran más cercanos a la región factible. Si $|A_{t+1}| > N$, se remueven los individuos con base en la vecindad del vecindario, este proceso se hace en forma iterativa hasta que se llega a N .

2.1.4. Algoritmo Implementado

A continuación se presenta el algoritmo adaptado para problemas multi-objetivo bajo incertidumbre:

2.2. MULTI-OBJECTIVE INTERVAL-VALUED PARTICLE SWARM OPTIMIZATION - MOIPSO37

Algorithm 2.1.1: SPEA2I(N : TamannodelaPoblacion, T : NumerodeGeneraciones)

Generar poblacion inicial P_0

$A_0 \leftarrow \Phi$

for $t \leftarrow 0$ **to** T

Calcular $S(i) \forall i \in P_t \cup A_t$

Calcular $R(i) \forall i \in P_t \cup A_t$

Calcular $D(i) \forall i \in P_t \cup A_t$

$F(i) \leftarrow R(i) + D(i) \forall i \in P_t \cup A_t$

$A_{t+1} \leftarrow \{i | i \in P_t \cup A_t \wedge F(i) < 1 \wedge |\{g_j(i) | (g_j(i) \not\leq 0)\}| = 0\}$

if $|A_{t+1}| < N$

then *Completar la poblacion con los $|A_{t+1}| - N$ mejores individuos dominados de $P_t \cup A_t$ que no violan restricciones*

do **if** $|A_{t+1}| < N$

then *Completar la poblacion con los $|A_{t+1}| - N$ mejores individuos dominados de $P_t \cup A_t$ que violan restricciones cercanos region factible*

else *Aplicar operador de truncamiento con base al estimador de densidad*

Aplicar operador de seleccion de ruleta para apareamiento

$P_{t+1} \leftarrow r_{\Theta_r^{(i)}}^{(t)}(P_t)$

$P_{t+1} \leftarrow m_{\Theta_m^{(i)}}^{(t)}(P_{t+1})$

return $(A)_t$

2.2. Multi-Objective Interval-Valued Particle Swarm Optimization - MOIPSO

Este algoritmo es basado en el MOPSO, se modificó introduciendo un operador mutación para evitar que el comportamiento social lleve a que las partículas tiendan a un punto y no exploren todo el espacio de búsqueda. Con este cambio el algoritmo quedaría de la siguiente forma:

Algorithm 2.2.1: MOIPSO()

```

for  $i \leftarrow 1$  to  $M - 1$ 
    comment: Inicializa aleatoriamente la partícula  $i$ 
     $X_0(i) \leftarrow random$ 
    comment: Inicializa la velocidad de la partícula  $i$ 
     $V_0(i) \leftarrow 0$ 
do { comment: Inicializa la mejor posición de la partícula  $i$ 
     $X_{lb}(i) \leftarrow X_0(i)$ 
    comment: Selecciona la mejor posición global
    if  $f(X_0(i)) < X_{gb}$ 
    then  $X_{gb} \leftarrow f(X_0(i))$ 
while que no se cumpla el criterio de parada
     $V_{t+1}(i) \leftarrow V_t(i) + C_1 \times r_1 \times (X_{lb}(i) - X_t(i)) + C_2 \times r_2 \times (X_{gb} - X_t(i))$ 
     $X_{t+1}(i) \leftarrow X_t(i) + V_t(i)$  Actualiza la posición de cada partícula
    if  $f(X_t(i)) < f(X_{lb}(i))$ 
    then  $X_{lb}(i) \leftarrow f(X_t(i))$  Evalua la función objetivo en cada partícula
do { if  $X_{lb}(i) < X_{gb}$ 
    then  $X_{gb} \leftarrow X_{lb}(i)$  Actualiza el mejor global
    comment: Operador de mutación
     $X_{t+1} \leftarrow m_{\Theta_m}^{(i)}(X_{t+1})$ 

```

2.3. Ejemplos Numéricos

Los ejemplos numéricos presentados en el capítulo anterior fueron utilizados con ambos algoritmos, los resultados se presentan en el cuadro 2.1, donde se puede observar que los resultados son muy cercanos a los obtenidos analíticamente.

Ejemplo	<i>Resultado Analítico</i>	<i>Resultado SPEA2I</i>	<i>Resultado MOPSOI</i>
1.2.1	$x = 0$	$x = 0,00153$	$x = 0$
1.2.2	$(x_1, x_2) = (1,8, 0,6)$	$(x_1, x_2) = (1,807, 0,592)$	$(x_1, x_2) = (1,802, 0,604)$

Cuadro 2.1: Resultados obtenidos con los algoritmos propuestos para los ejemplos del capítulo 1

También se verificaron los algoritmos con varias funciones que se utilizan para probar los algoritmos de optimización, los cuales se presentan a continuación:

Ejemplo 2.3.1. Función de Rastrigin: Es una función no convexa, esta función es un problema bastante difícil debido a su gran espacio de búsqueda y su gran número de mínimos locales, ver figura

2.1. El problema de optimización esta definido por:

$$\begin{aligned} \text{mín } f(x_1, x_2) &= \left[19 + x_1^2 + x_2^2 - 10 * ((\cos(2\pi x_1) + \cos(2\pi x_2))), 21 + x_1^2 + x_2^2 - 10 * ((\cos(2\pi x_1) + \cos(2\pi x_2))) \right] \\ \text{s.a. } & -5 \leq x_1 \leq 5 \\ & -5 \leq x_2 \leq 5 \end{aligned} \quad (2.3)$$

Esta función tiene un mínimo global en $(x_1, x_2) = (0, 0)$ cuando $f(x) = [-1, 1]$. Resolviéndolo con los algoritmos propuestos se obtiene que la solución con el SPEA2I es $(x_1, x_2) = (0,002, 0,001)$ y con el MOPSOI es $(x_1, x_2) = (0,006, 0,001)$.

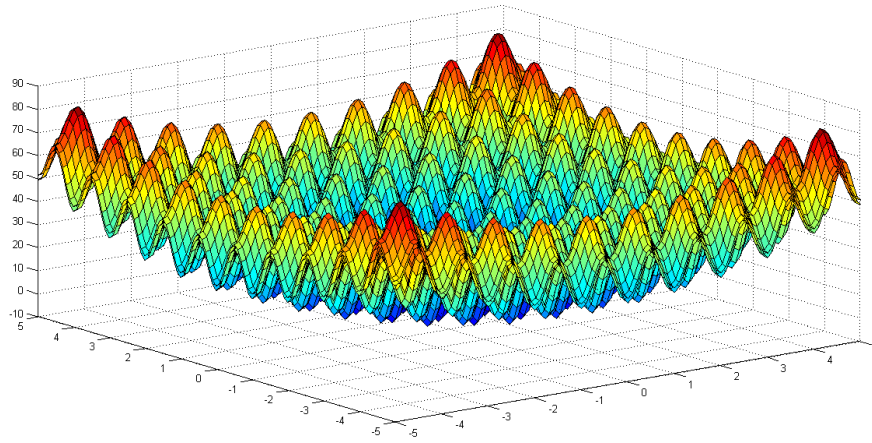


Figura 2.1: Función de Rastrigin

Ejemplo 2.3.2. Función de Rosenbrock: Es una función no convexa. El mínimo global está dentro de valle largo, estrecho y en forma parabólica, ver figura 2.2. El problema de optimización esta definido por:

$$\begin{aligned} \text{mín } f(x_1, x_2) &= \left[(1 - x_1)^2 + 100 * (x_2 - x_1^2)^2, 1 + (1 - x_1)^2 + 100 * (x_2 - x_1^2)^2 \right] \\ \text{s.a. } & -1 \leq x_1 \leq 3 \\ & -1 \leq x_2 \leq 3 \end{aligned} \quad (2.4)$$

Esta función tiene un mínimo global en $(x_1, x_2) = (1, 1)$ cuando $f(x) = [0, 1]$. Resolviéndolo con los algoritmos propuestos se obtiene que la solución con el SPEA2I es $(x_1, x_2) = (0,962, 0,975)$ y con el MOPSOI es $(x_1, x_2) = (1,001, 1,002)$.

Ejemplo 2.3.3. Función de Schwefel: Es engañosa ya que el mínimo global es geoméricamente distante al mejor mínimo local, ver 2.3. El problema de optimización esta definido por:

$$\begin{aligned} \text{mín } f(x_1, x_2) &= \left[837,9658 - x_1 \sin(\sqrt{x_1}) - x_2 \sin(\sqrt{x_2}), 838,9658 - x_1 \sin(\sqrt{x_1}) - x_2 \sin(\sqrt{x_2}) \right] \\ \text{s.a. } & -500 \leq x_1 \leq 500 \\ & -500 \leq x_2 \leq 500 \end{aligned} \quad (2.5)$$

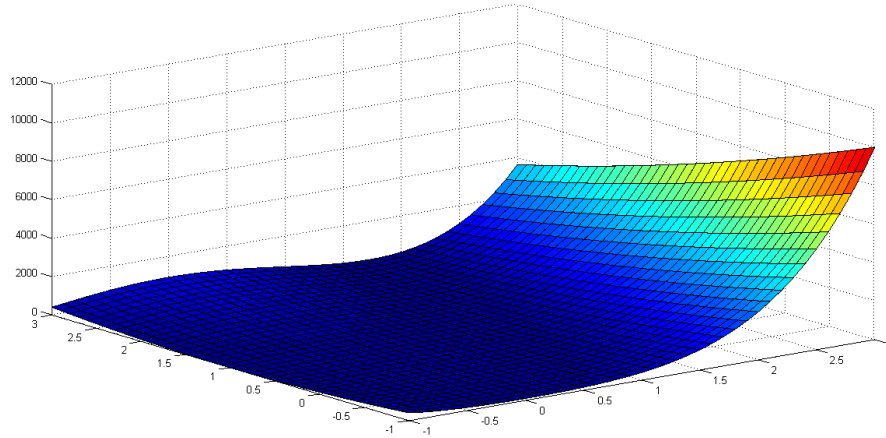


Figura 2.2: Función de Rosenbrock

Su mínimo global es $f(x) = [0, 1]$ y es obtenido para $(x_1, x_2) = (420,9687, 420,9687)$. Resolviéndolo con los algoritmos propuestos se obtiene que la solución con el SPEA2I es $(x_1, x_2) = (421,253, 420,799)$ y con el MOPSOI es $(x_1, x_2) = (420,9624, 420,9774)$.

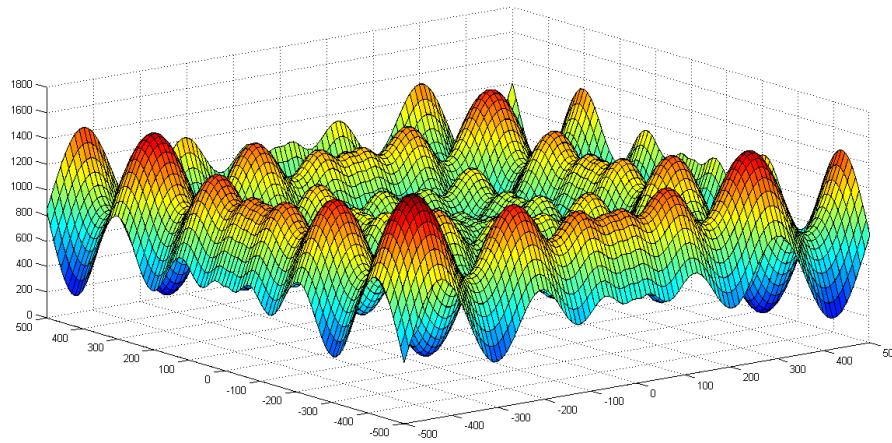


Figura 2.3: Función de Schwefel

Ejemplo 2.3.4. Función de Michalewicz: La función está definida por:

$$\text{mín } f(x_1, x_2) = \left[837,9658 - x_1 \sin(\sqrt{x_1}) - x_2 \sin(\sqrt{x_2}), 838,9658 - x_1 \sin(\sqrt{x_1}) - x_2 \sin(\sqrt{x_2}) \right]$$

$$\text{s.a. } 0 \leq x_1 \leq \pi$$

$$0 \leq x_2 \leq \pi$$

(2.6)

Su mínimo global es $f(x) = [-2, 0]$ y es obtenido para $(x_1, x_2) = (\frac{\pi}{2}, \frac{\pi}{2})$, ver 2.4. Al resolverlo con

algoritmos propuestos se obtiene que la solución con ambos algoritmos es $(x_1, x_2) = (1,571, 1,571)$.

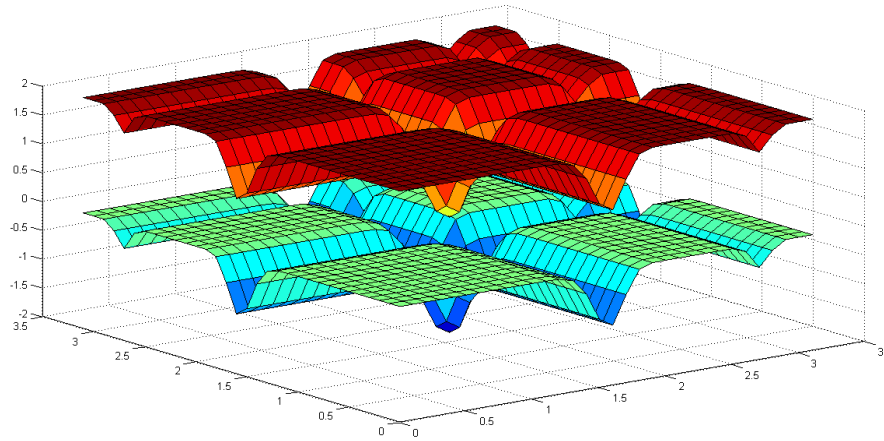


Figura 2.4: Función de Michalewicz

Capítulo 3

Problema Inverso

En este trabajo se ha incluido un capítulo sobre problemas inversos, dado que este tipo de problemas puede ser visto desde la perspectiva de la optimización bajo incertidumbre, ya que su objetivo es estimar parámetros a través de datos observados los cuales son inciertos por imprecisión al momento de ser medidos.

3.1. Conceptos Básicos

En esta sección se explicarán los conceptos básicos del Problema Inverso: qué es, cómo se modela, entre otros.

Definición 3.1.1. (*Problema Directo*): Se define como problema directo aquel que predice valores libres de errores de los parámetros observables d , los cuales corresponden al modelo dado m

$$m \rightarrow d = g(m) \quad (3.1)$$

donde g es denominado operador directo, g expresa el modelo matemático.

Definición 3.1.2. (*Problema Inverso*) Un Problema Inverso es aquel en donde los valores de algunos parámetros del modelo deben ser obtenidos de los datos observados.

Los problemas inversos normalmente son problemas mal planteados. De las tres condiciones de un problema bien planteado (existencia, unicidad, estabilidad de la solución o soluciones) las condiciones de estabilidad y unicidad son las que más a menudo se quebrantan. El modelo del Problema Inverso es el siguiente:

$$d = g(m) \quad (3.2)$$

donde g representa el modelo

Definición 3.1.3. (*Información A Priori*): Se considera información a priori aquella que es obtenida independientemente de los resultados medidos. La función de densidad de probabilidad de la información a priori se denota por $\rho_M(m)$.

Existen dos posiciones en cuanto a utilizar información a priori:

1. No se debe utilizar información a priori porque sesga los resultados.
2. Desde la perspectiva bayesiana, si se debe utilizar la información *a priori* porque puede ayudar a realizar un mejor ajuste de los parámetros con la información *a posteriori*.

En este trabajo, se asume la segunda posición. Uno de los autores más reconocidos en esta dirección es Tarantola y en su artículo "Bayes and the inverse problem"[27] sugiere que para la solución de un Problema Inverso se debe utilizar la información *a priori* para crear secuencialmente modelos del sistema, los cuales pueden ser infinitos. Luego, para cada modelo se debe resolver el problema directo y comparar los datos predichos con los observados, teniendo un criterio para determinar si el modelo es aceptable o inaceptable, de acuerdo a la incertidumbre de las observaciones y la teoría física. Así, los modelos no aceptados son falseados y deben ser rechazados. El conjunto de modelos que no son falseados son las soluciones del Problema Inverso (no unicidad de la solución).

Para configurar de mejor manera esta perspectiva del Problema Inverso, se requieren las siguientes definiciones:

Definición 3.1.4. (*Espacio de Datos*): Se define como el espacio de datos \mathbb{D} , el espacio de todas las posibles respuestas de los instrumentos de medición.

Definición 3.1.5. (*Espacio del Modelo*): Una selección particular de parámetros del modelo es una parametrización del sistema, cada parametrización del sistema; puede ser representada como un punto de un espacio abstracto denominado espacio del modelo \mathbb{M} .

Definición 3.1.6. (*Función de Densidad de Probabilidad de los datos*): Es una función sobre el Espacio de Datos \mathbb{D} , estos constan de incertidumbre, entonces si $d = \{d_1 \dots, d_n\}$ representan el conjunto de parámetros observados, el resultado de la medición es representado a través de la función probabilidad de densidad $\rho_{\mathbb{D}}(d)$ sobre el espacio de datos \mathbb{D} .

Definición 3.1.7. (*Función de Densidad de Probabilidad Conjunta*): La función de densidad de probabilidad conjunta de los datos (d) y parámetros del modelo (m) es la distribución de la intersección de eventos de d y m ; como son independientes entre sí, la función de probabilidad de densidad conjunta es definida como el producto de las funciones densidad de probabilidad de cada uno, así: $\rho(d, m) = \rho(d)\rho(m)$

Definición 3.1.8. : La probabilidad de Densidad Teórica $\Theta(d, m)$ representa la información sobre la correlaciones, obtenidas de una ley física, entre d y m

Definición 3.1.9. (*Probabilidad de Densidad $\sigma(m)$*): Representa la información a posteriori del modelo, deducida de $\rho(m)$ y el grado de ajuste entre los datos predichos del modelo y los datos observados. $\sigma(m) = k \rho(m)L(m)$, donde k es la constante de normalización

3.2. Criterios de Identificación

El Problema Inverso se reduce a encontrar un conjunto de parámetros m , tales que las variables calculadas con dichos parámetros d , a través del modelo, se ajusten lo mejor posible a las medidas realizadas d^* .

Este *ajuste* se realiza matemáticamente a través de un criterio de identificación. Los criterios de identificación que permiten determinar el *ajuste* (función objetivo) usualmente son:

- Minimizar distancias
- Markov
- Maximizar la verosimilitud

Cada uno de los métodos de identificación siguen una hipótesis acerca de la distribución de las observaciones:

- La distribución Laplaciana $f(x) \exp(-|x|)$ en el método del menor valor absoluto
- La distribución Gaussiana $f(x) \exp(-x^2)$ en el método de mínimos cuadrados

3.2.1. Minimizar Distancias

Es el criterio más simple, consiste en minimizar la distancia entre los datos observados y los datos estimados con el modelo (error de predicción). Se puede utilizar la distancia Euclideana o generalizarlo a la norma de orden j :

- Norma 1 $\|g(m) - d_{obs}\|_1 = \sum |g(m)^i - d_{obs}^i|$
- Norma 2 $\|g(m) - d_{obs}\|_2 = \left[\sum |g(m)^i - d_{obs}^i|^2 \right]^{\frac{1}{2}}$
...
- Norma j $\|g(m) - d_{obs}\|_j = \left[\sum |g(m)^i - d_{obs}^i|^j \right]^{\frac{1}{j}}$

A medida que se aumenta el orden de la norma se da más importancia a los errores más grandes; el límite de esta norma es:

$$\|g(m) - d_{obs}\|_{\infty} = \text{máx} |g(m)^i - d_{obs}^i|$$

De estas normas, las más utilizadas son minimizar la suma de los valores absolutos de los errores y minimizar la suma la raíz cuadrada de los errores (mínimos cuadrados); este último se puede representar en forma matricial de la siguiente forma:

$$[g(m) - d_{obs}]^T [g(m) - d_{obs}]$$

3.2.2. Markov

Es una generalización del criterio anterior, donde la función objetivo es:

$$\min_{m \in M} [g(m) - d_{obs}]^T W [g(m) - d_{obs}] \quad (3.3)$$

donde W es una matriz de pesos que se asigna con base en la información a priori, otorgando mayor peso a las observaciones más fiables.

3.2.3. Maximizar Verosimilitud

La diferencia entre los datos observados y los datos predichos por el modelo se considera un error, el cual se define como aleatorio, porque se produce por eventos únicos imposibles de controlar durante el proceso de medición; por lo tanto tiene asociada una distribución de probabilidad y se puede utilizar un criterio estadístico para la identificación del modelo, concretamente la máxima verosimilitud. Se define la máxima verosimilitud como la medida del grado de ajuste entre los datos predichos por el modelo y los datos observados. Este criterio requiere mayor información relacionada con la fiabilidad de los datos observados: la función de probabilidad y la correlación de los datos.

Cuando los datos observados tiene una distribución Gaussiana, la máxima verosimilitud se expresa de la siguiente forma:

$$L(m) = k \exp\left(\frac{1}{2}(g(m) - d_{obs})^T C^{-1}(g(m) - d_{obs})\right) \quad (3.4)$$

Por comodidad al momento de realizar la optimización no se trabaja con la función de la verosimilitud, si no con la función de soporte, la cual se define como:

$$S(m) = \ln L(m) \quad (3.5)$$

Resolviendo la ecuación cuando los datos observados tienen distribución Gaussiana se obtiene:

$$S(m) = \ln k + \frac{1}{2}(g(m) - d_{obs})^T C^{-1}(g(m) - d_{obs}) \quad (3.6)$$

Para efectos de optimización es equivalente maximizar $L(m)$ a minimizar $S(m)$, por lo tanto para el caso de la distribución normal la función objetivo a minimizar sería:

$$\text{mín}(g(m) - d_{obs})^T C^{-1}(g(m) - d_{obs}) \quad (3.7)$$

Esta ecuación tiene una forma similar a la función utilizada en el criterio Markov; pero aquí la matriz de pesos tiene un significado estadístico, es la inversa de la matriz de covarianzas de los datos observados. Note que si esta matriz es diagonal entonces el criterio de identificación será Markov y si es la matriz identidad el criterio es minimizar distancias, este último caso se presenta cuando los datos observados son independientes y la varianza es igual para todos.

Existe una variación del criterio de máxima verosimilitud cuando se tiene información *a priori*, así la función objetivo es modificada para introducir esta información, a la cual se le incorpora la función de densidad de los datos observados. Cuando los datos tienen una distribución Gaussiana, la función de soporte quedaría de la siguiente forma:

$$S(m) = \frac{1}{2}(g(m) - d_{obs})^T C_D^{-1}(g(m) - d_{obs}) + (m - m_{prior})^T C_M^{-1}(m - m_{prior}) \quad (3.8)$$

donde C_D es la matriz de covarianzas de los datos observados y C_M la matriz de covarianzas del modelo con base en la información *a priori*.

Al introducir esta información en la función a optimizar se le está dando un enfoque bayesiano al criterio de identificación.

3.3. Métodos de Solución

Existen diferentes métodos numéricos y heurísticos para solucionar el Problema Inverso, a continuación se presentarán los más representativos:

3.3.1. Métodos Numéricos

3.3.1.1. Gauss - Newton (GNA)

Se utiliza para resolver problemas no lineales de mínimos cuadrados. Es una modificación del método de optimización de Newton que no usa segundas derivadas. Requiere que f sea diferenciable y que f' satisfaga las condiciones de Lipschitz [3].

3.3.1.2. Levenberg - Marquardt (LMA)

Interpola entre el Algoritmo de Gauss - Newton (GNA) y el método de descenso de gradiente. El LMA es más robusto que el GNA [9].

3.3.2. Métodos Heurísticos

- Metropolis: Es un método de Monte Carlo Cadena de Markov, es aleatorio y no tiene memoria. Cada paso sólo depende del paso anterior.
- Recocido Simulado (Simulated Annealing - SA -): Es una adaptación del algoritmo de Metropolis. En cada iteración evalúa algunos vecinos del estado actual y probabilísticamente decide entre cambiar el sistema al estado nuevo o quedarse en el estado actual. A continuación se presenta el pseudocódigo de este método:

Algorithm 3.3.1: SIMULATEDANNEALING($T_0, \alpha, \epsilon, P_0$)

```

 $P_{current} \leftarrow P_0$  , Solucion inicial
 $T \leftarrow T_0$  , Temperatura inicial
 $P_{best} \leftarrow P_{current}$ 
while  $T \geq \epsilon$ 
  do {
     $P_{new} \leftarrow neighbour(P_{current})$ 
    if  $f(P_{new}) < f(P_{best})$ 
      then {
         $P_{best} \leftarrow P_{new}$ 
         $P_{current} \leftarrow P_{new}$ 
      }
    else if  $e^{-\Delta/T} > random$ 
      then {
         $P_{current} \leftarrow P_{new}$ 
      }
     $T \leftarrow \alpha T$ 
  }
return  $(P)_{best}$ 

```

- Evolutivos (EA)
 - Método práctico para la solución de problemas multi-objetivo
 - Se aproximan a la frontera de Pareto
 - Tratan de minimizar la distancia a la frontera de Pareto y maximizar la diversidad de las soluciones, a través de:

- Asignación del ajuste
- Preservación de la diversidad
- Elitismo

En el capítulo 2 se presenta mayor detalle sobre este tipo de algoritmos.

3.4. Ejemplos Numéricos

En esta sección se presentan algunos problemas inversos los cuáles fueron solucionados utilizando el algoritmo evolutivo SPEA2 descrito en el capítulo anterior.

3.4.1. Estimación de las Coordenadas del Epicentro de un Evento Sísmico

Un evento sísmico inicia en el momento $T = 0$ en una ubicación desconocida de la superficie de la tierra. Las ondas sísmicas producidas por la explosión han sido registradas por seis estaciones sísmicas, las cuales están ubicadas en un sistema en las siguientes coordenadas rectangulares:

$$\begin{aligned}
 (x_1, y_1) &= (3km, 15km) , (x_2, y_2) = (3km, 16km), \\
 (x_3, y_3) &= (4km, 15km) , (x_4, y_4) = (4km, 16km), \\
 (x_5, y_5) &= (5km, 15km) , (x_6, y_6) = (5km, 16km).
 \end{aligned}
 \tag{3.9}$$

Los tiempos de llegadas de las ondas sísmicas a estas estaciones son:

$$\begin{aligned}
 t_{obs}^1 &= 3,12 \pm \sigma , t_{obs}^2 = 3,26 \pm \sigma , \\
 t_{obs}^3 &= 2,98 \pm \sigma , t_{obs}^4 = 3,12 \pm \sigma , \\
 t_{obs}^5 &= 2,84 \pm \sigma , t_{obs}^6 = 2,98 \pm \sigma ,
 \end{aligned}
 \tag{3.10}$$

donde $\sigma = 0,10s$, el símbolo $\pm\sigma$ es una notación corta que indica la incertidumbre de los datos, los cuales son independientes y pueden ser modelados a través de una distribución Gaussiana con una desviación estándar σ .

Se requiere estimar la coordenadas (X, Y) del epicentro de la explosión, asumiendo que la velocidad de las ondas sísmicas es $v = 5km/s$. Suponga una aproximación a la superficie de la Tierra plana y considere que las coordenadas de las estaciones son cartesianas.

El problema fue resuelto con los siguientes parámetros:

- Tamaño población: 100
- Tamaño de la descendencia: 80
- Número de Generaciones: 100
- Probabilidad de Mutación: 0.05

- Espacio de búsqueda:
 $X \in [0, 20]$
 $Y \in [0, 20]$

El resultado promedio en cada generación obtenido se presenta en la figura 3.1 y los diez mejores individuos en el cuadro 3.1

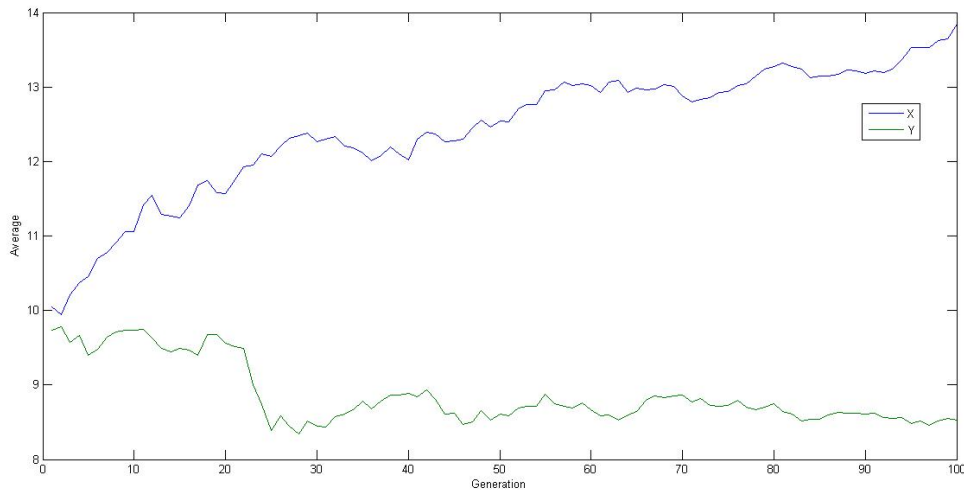


Figura 3.1: Valor Promedio de las Coordenadas de Epicentro

Cuadro 3.1: Diez mejores resultados de las Coordenadas del Epicentro

X	Y
14.6074	5.9130
14.6611	8.9292
14.7044	7.7238
14.7324	7.0764
14.7331	7.5231
14.8422	9.4842
14.8659	6.5749
14.8659	7.3930
15.0209	8.8197
15.0557	6.4920

3.4.2. Medición de la Aceleración Gravitacional

Un gravímetro absoluto se utiliza para medir el valor de la aceleración gravitacional g de una masa cuando cae en el vacío. Una masa es lanzada hacia arriba con una velocidad inicial v_0 , se

tomó la medida de la alturas z_i que tenía la masa en el instante instantes t_i , donde el modelo teórico del movimiento parabólico es:

$$z(t) = v_0 t - \frac{1}{2} g t^2 \quad (3.11)$$

Los valores medidos de z_i y t_i deben ser utilizados para inferir los valores de v_0 y g . Las mediciones realizadas durante el experimento de caída libre fueron:

$$\begin{aligned} t_1 &= 0,20s \pm 0,01s, \quad z_1 = 0,62m \pm 0,02m, \\ t_2 &= 0,40s \pm 0,01s, \quad z_2 = 0,88m \pm 0,02m, \\ t_3 &= 0,60s \pm 0,01s, \quad z_3 = 0,70m \pm 0,02m, \\ t_4 &= 0,80s \pm 0,01s, \quad z_4 = 0,15m \pm 0,02m, \end{aligned} \quad (3.12)$$

El algoritmo SPEA2 fue ejecutado con los siguientes parámetros de entrada:

- Tamaño población: 100
- Tamaño de la descendencia: 100
- Número de Generaciones: 100
- Probabilidad de Mutación: 0.05
- Espacio de búsqueda:
 $X \in [0, 5]$
 $Y \in [10, 15]$

El resultado promedio obtenido en cada generación se presenta en la figura 3.2 y los diez mejores resultados son presentados en el cuadro 3.2

Cuadro 3.2: Diez mejores resultados para la Velocidad Inicial y la Gravedad

Velocity	Gravity
3.3630	9.7544
3.3640	9.7574
3.3864	9.7584
3.3902	9.7594
3.7142	9.7594
3.7511	9.7594
3.7529	9.7600
3.7869	9.7605
4.0507	9.7607
4.1746	9.7607

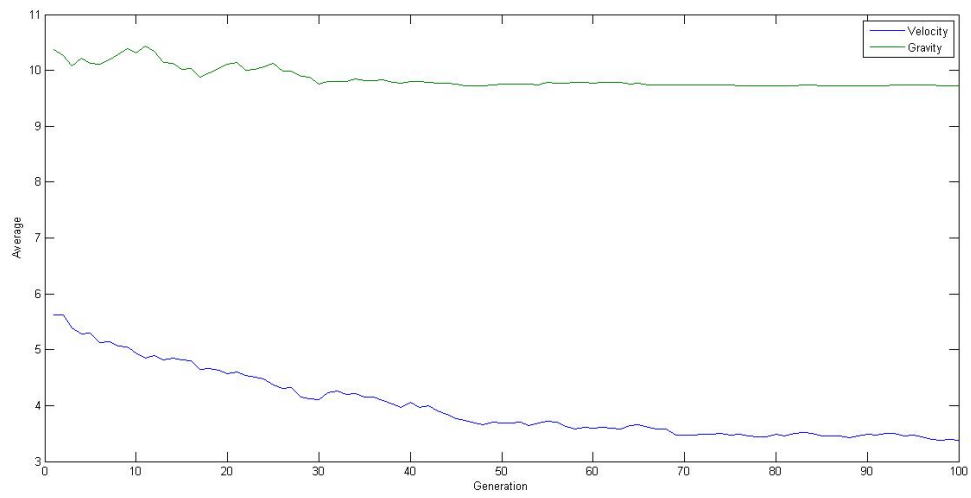


Figura 3.2: Valores Promedio de la Velocidad Inicial y Gravedad

Capítulo 4

Aplicaciones

En esta sección se presentará una aplicación de los capítulos 2 y 3 para ajustar los parámetros del modelo logístico, aplicado para proyectar la población de la ciudad de Medellín, con base en los datos obtenidos en los censos realizados en el último siglo. Sobre este modelo, en particular, ya se han realizado trabajos anteriores como es [18]. En primera instancia, se exponen los elementos básicos del modelo logístico.

4.1. Modelo Logístico

El modelo logístico es un modelo de crecimiento poblacional que se soporta en las siguiente características:

1. El régimen de crecimiento logístico es válido cuando se supone que la dinámica de crecimiento en el tiempo, está determinada por tres hipótesis básicas, a saber:
 - a) El crecimiento vegetativo (más las inmigraciones, si existen), es en todo momento proporcional al número de individuos presentes, el cual se denomina p .
 - b) La población vive en un espacio limitado o bajo condiciones externas restrictivas que no le permitirán, ni aún a largo plazo, exceder de un tamaño límite, el cual se denomina L .
 - c) En todo momento la velocidad de crecimiento es proporcional al factor $(L - p)$ que representa las fuerzas exógenas restrictivas.

Dentro de las hipótesis anteriores, y si se sabe que el dato censal, que es históricamente más reciente, corresponde al valor más alto ya censado, se determina que la ecuación diferencial para el crecimiento de una población ascendentemente creciente, es:

$$\frac{dp}{dt} = bp(L - p) \quad (4.1)$$

donde:

p : Número de individuos de la población que viven en el instante t del tiempo.

b : Coeficiente constante y positivo que es propio de las características vitales de la población y

del régimen biológico y ecológico en que vive.

L : Población límite posible, que está determinada por el entorno vital de la población.

Integrando la ecuación diferencial (4.1)

$$\frac{dp}{p(L-p)} = bdt \quad (4.2)$$

$$\int \frac{dp}{p(L-p)} = b \int dt \quad (4.3)$$

$$\frac{1}{L} \left(\int \frac{dp}{p} + \int \frac{dp}{L-p} \right) = b \int dt \quad (4.4)$$

se obtiene la función de crecimiento de la población:

$$\hat{p}(t) = \frac{L}{1 + Ke^{-bt}} \quad (4.5)$$

en donde K es una constante que se introduce al efectuar la integración y que, numéricamente, depende del volumen que tenga la población en el momento que se tome como instante inicial ($t = 0$) para comenzar a contar el tiempo. En efecto, de la ecuación (4.5) se deduce el valor de K , de la siguiente forma:

$$p(0) = \frac{L}{1 + Ke^{-b(0)}} \quad (4.6)$$

$$p(0) = \frac{L}{1 + K} \quad (4.7)$$

$$K = \frac{L}{p(0)} - 1 \quad (4.8)$$

Se puede observar que la ecuación (4.5) es invariante ante un cambio en el origen de tiempo o unidad de medida del tiempo.

2. En las aplicaciones demográficas, se trata usualmente de *ajustar* una función de este tipo a una sucesión consecutiva de valores numéricos de la población p_1, p_2, \dots, p_n , los cuales son medidos experimentalmente (por ejemplo, mediante recuentos o censos de la población, en momentos sucesivos y distintos del tiempo t_1, t_2, \dots, t_n , los cuales se disponen en orden creciente, pero no son necesariamente equidistantes).

Estos datos presentan tres características prácticas, que son importantes para entender el problema teórico del ajuste de la curva logística:

- a) Casi inevitablemente están afectados por errores de recuento (por ejemplo sobre-enumeración, sub-enumeración) o pueden ser el resultado de estimaciones y no de enumeraciones precisas.
- b) Aunque tales datos fueran rigurosamente exactos, no puede esperarse que todos caigan rigurosa y exactamente sobre una función basada en hipótesis determinísticas (como lo es la logística) ya que en el crecimiento de un agregado colectivo, como en el caso de una población biológica, hay inevitablemente desviaciones aleatorias más o menos apreciables con respecto a la función determinística.

- c) Sucede entonces que, aun cuando la realidad geográfica, ambiental, ecológica y social justifique la adopción de una curva logística de crecimiento, el problema del ajuste a los datos empíricamente observados en el tiempo, presenta varias dificultades de técnica numérica. Ese problema consiste, en esencia, en calcular los tres parámetros L , K , b de la ecuación (4.5), a partir de la serie de recuentos, que suele darse en forma tabulada:

Fecha	Población
t_1	p_1
t_2	p_2
...	...
t_n	p_n

3. La tasa relativa de crecimiento poblacional es una variable muy importante en todos los estudios demográficos y está definida por la identidad:

$$r(t) = \frac{1}{p} \frac{dp}{dt} \quad (4.9)$$

que se interpreta como la velocidad del crecimiento de la población, dividida por la población. Al efectuar el proceso de derivación correspondiente, se obtiene:

$$\frac{dp}{dt} = \frac{d}{dt} \left(\frac{L}{1 + Ke^{-Bt}} \right) \quad (4.10)$$

$$\frac{dp}{dt} = \frac{LKbe^{-Bt}}{(1 + Ke^{-Bt})^2} \quad (4.11)$$

reemplazando en la ecuación (4.9) los valores para p y $\frac{dp}{dt}$ se obtiene:

$$r(t) = \left(\frac{1 + Ke^{-Bt}}{L} \right) \left(\frac{LKbe^{-Bt}}{(1 + Ke^{-Bt})^2} \right) \quad (4.12)$$

$$r(t) = \frac{Kbe^{-Bt}}{1 + Ke^{-Bt}} \quad (4.13)$$

$$r(t) = \frac{Kb}{K + e^{Bt}} \quad (4.14)$$

Generalmente el año cero se puede tomar como un año intermedio en la serie de fechas de los datos censales.

4.1.1. Aplicación del Modelo Logístico a los Censos de Medellín

A partir de los datos de los censos oficiales que se han hecho desde 1912 hasta 2005, suministrados por el Departamento Nacional de Estadística - DANE - (cuadro 4.1) y utilizando el modelo logístico se planteo el siguiente problema de optimización

$$\text{mín } f(t) = \left| p(t) - \frac{L}{1 + Ke^{-Bt}} \right| \quad (4.15)$$

Fecha	Población ($p(t_i)$)	t_i
Marzo 5 de 1912	71004	0,177
Octubre 14 de 1918	79146	6,785
Noviembre 17 de 1928	120044	16,876
Julio 5 de 1938	168266	26,511
Mayo 9 de 1951	358189	39,355
Julio 15 de 1964	791589	52,538
Octubre 24 de 1973	1163865	61,8118
Octubre 15 de 1985	1480382	73,788
Octubre 15 de 1993	1834881	81,788
Mayo 22 de 2005	2223078	93,390

Cuadro 4.1: Datos de población para la ciudad de Medellín. t_i : Año censal contado a partir del año base (1912)

Como puede observarse este problema se encuentra dentro de la categoría de los problemas inversos, donde se deben estimar los parámetros L , K y B , y considerar la incertidumbre de los datos observados, pues ellos no son datos precisos. Para resolver este problema se modificó la función objetivo, donde no se minimiza la distancia entre los datos predichos por el modelo y los datos observados, sino que se utiliza el máximo error relativo, dado que el error absoluto no proporciona suficiente información, porque los valores poblacionales se encuentran en escalas muy diferentes.

Por lo tanto, el problema a optimizar es:

$$\text{mín } f(t) = \text{máx } \left| \frac{p(t) - \frac{L}{1+Ke^{-Bt}}}{p(t)} \right| \quad (4.16)$$

Note que como los valores obtenidos en un censo poblacional no son exactos, es decir, son imprecisos, estos tienen un error esperado, por lo tanto cada uno de los datos poblacionales se encuentra dentro de un intervalo, ocasionando que la función $p(t) - \frac{L}{1+Ke^{-Bt}}$ sea intervalo-valuada. El algoritmo evolutivo SPEA2 se utilizó para resolver este problema, se ejecutó diez veces con los siguientes parámetros:

- Tamaño población: 100
- Tamaño de la descendencia: 100
- Número de Generaciones: 200
- Probabilidad de Mutación: 0.05
- Espacio de búsqueda:
 - $L \in [3400000, 3500000]$
 - $K \in [55, 60]$
 - $b \in [0,04, 0,1]$

En la gráfica 4.1 se presenta una curva de la población censada y la población estimada con los diferentes parámetros observados en el cuadro 4.1.1.

Resultado	f(t)	L	K	b
Mejor	20,493 %	3404198,17	59,84	0,0503
Promedio	20,956 %	3427449,81	59,49	0,0499
Peor	21,4962 %	3414791,19	58,32	0,0494

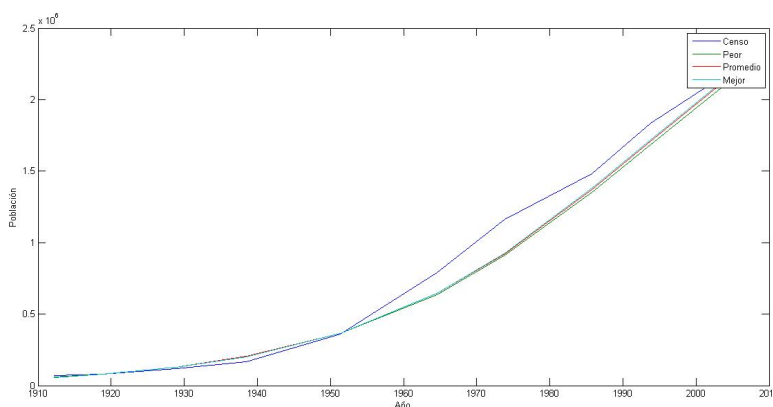


Figura 4.1: Modelo Logístico: Comparación de los Datos Observados y los Datos Predichos

En el cuadro 4.2 se comparan los valores históricos de los censos con la estimación que da la curva logística generada con base en los resultados anteriores para el mejor caso y también se incluyen los datos estimados por la curva logística estimada en [18], para los diferentes años desde 1912. Esta comparación también se presenta gráficamente en 4.2.

Fecha	t_i	Pbn. Censada	Pbn. Estimada Modelo	Pbn. Estimada en [18]
Marzo 5 de 1912	0,177	71004	56445	58135
Octubre 14 de 1918	6,785	79146	78165	81058
Noviembre 17 de 1928	16,876	120044	127857	133938
Julio 5 de 1938	26,511	168266	202749	214218
Mayo 9 de 1951	39,355	358189	366810	391384
Julio 15 de 1964	52,538	791589	646068	693703
Octubre 24 de 1973	61,8118	1163865	925384	994469
Octubre 15 de 1985	73,788	1480382	1379680	1476559
Octubre 15 de 1993	81,788	1834881	1717896	1828081
Mayo 22 de 2005	93,390	2223078	2199211,83	2316083

Cuadro 4.2: Datos de comparación entre los datos del censo, los datos estimados por el modelo y los datos estimados en [18]

Además, en la tabla 4.3 se compararon las proyecciones realizadas para la población de Medellín por el DANE para los años 2006 al 2020 con los datos proyectados por el modelo y en [18].

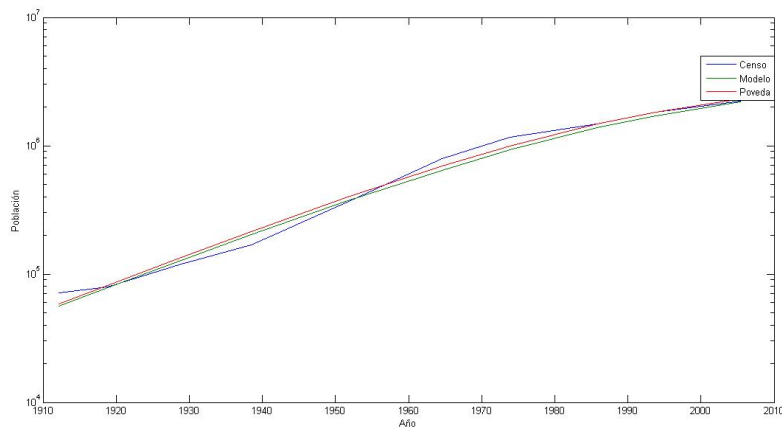


Figura 4.2: Modelo Logístico: Comparación de los Datos del Censo, los Datos Predichos por el Modelo y los Datos Predichos en [18]

4.1.2. Aplicación del Modelo Logístico a los Datos de Rangos de Edad para Antioquia

También se resolvió el modelo logístico para los datos poblacionales de Antioquia en cada uno de los grupos quinquenales desde los 0 años a los 29 años con base en los datos suministrados por el DANE desde 1918 hasta 2005 (tabla 4.4), utilizando el problema de optimización 4.15 para cada uno de los quinquenales.

Para resolverlo se utilizó el algoritmo SPEA2I con los siguientes parámetros:

- Tamaño población: 100
- Tamaño de la descendencia: 100
- Número de Generaciones: 200
- Probabilidad de Mutación: 0.05

El espacio de búsqueda definido para cada quinquenal se presenta en la tabla 4.5 y los datos estimados con el modelo en la tabla 4.6.

Año	t_i	Pbn. Estimada DANE	Pbn. Estimada Modelo	Pbn. Estimada en [18]
2006	94	2239003	2222977	2338853
2007	95	2265244	2261437	2376944
2008	96	2291378	2299267	2414298
2009	97	2317336	2336437	2450988
2010	98	2343049	2372917	2486789
2011	99	2368282	2408682	2521783
2012	100	2393011	2443708	2555952
2013	101	2417325	2477976	2589372
2014	102	2441123	2511469	2621848
2015	103	2464322	2544170	2653464
2016	104	2486723	2576068	2684214
2017	105	2508452	2607154	2714175
2018	106	2529403	2637421	2743182
2019	107	2549537	2666863	2771319
2020	108	2569007	2695478	2798589

Cuadro 4.3: Datos de comparación entre los datos del censo, los datos estimados por el modelo y los datos estimados en [18]

<i>Edad</i>	10/1918	11/1928	07/1938	05/1951	07/1964	10/1973	10/1985	10/1993	05/2005
< 1	19905	34502	40311	57127	93615	72787	85991	88254	107596
1 - 4	105224	126044	144539	207982	357151	330392	367823	386645	403297
5 - 9	109174	136436	172723	212249	398.433	454699	452919	485777	546948
10 - 14	98962	126787	150855	184954	326481	436486	419193	487364	571243
15 - 19	83146	108437	124621	160638	253130	362790	455298	405838	531578
20 - 24	72551	94262	108794	149386	195651	276097	450472	399223	48090
25 - 29	62357	81534	87490	111095	158316	201295	364183	409029	431231

Cuadro 4.4: Datos de población por quinquenal para el Departamento de Antioquia

<i>Quinquenal</i>	L	K	b
< 1	[130000, 180000]	[5, 10]	[0, 0,1]
1 - 4	[430000, 680000]	[5, 10]	[0, 0,1]
5 - 9	[600000, 800000]	[5, 10]	[0, 0,1]
10 - 14	[600000, 800000]	[5, 10]	[0, 0,1]
15 - 19	[500000, 700000]	[5, 10]	[0, 0,1]
20 - 24	[500000, 700000]	[5, 10]	[0, 0,1]
25 - 29	[500000, 700000]	[10, 15]	[0, 0,1]

Cuadro 4.5: Parámetros de ejecución para el SPEA2I en el problema de estimación de parámetros del modelo logístico a los datos de quinquenales para Antioquia

<i>Edad</i>	10/1918	11/1928	07/1938	05/1951	07/1964	10/1973	10/1985	10/1993	05/2005
< 1	23977	32488	42330	57531	74183	85464	98392	105651	114085
1 - 4	88426	124361	166680	232012	301425	346114	394148	419315	446618
5 - 9	92219	128351	172530	246277	335717	402137	484818	534628	595845
10 - 14	81420	113344	152471	218067	298175	358107	433292	478925	535390
15 - 19	70151	97628	131372	188167	257998	310636	377216	417953	468741
20 - 24	69307	94352	124715	175559	238675	287350	351081	391706	444703
25 - 29	51394	69983	92985	132982	185880	229684	291892	334981	396082

Cuadro 4.6: Datos de población por quinquenal para el Departamento de Antioquia obtenidos con el modelo

Conclusiones

Resolver un problema de optimización multi-objetivo bajo incertidumbre con restricciones a través de técnicas analíticas, requiere que las funciones involucradas en el modelo tengan condiciones analíticas como convexidad y diferenciabilidad, entre otras, lo cual no siempre se satisface en problemas reales; sin embargo, los métodos computacionales son más laxos y permiten dar soluciones a un espectro más grande de problemas.

En este trabajo se ha mostrado como los métodos computacionales son una buena alternativa al momento de resolver problemas multi-objetivo, porque generan soluciones cercanas a las frontera de Pareto, son robustos, tienen gran capacidad de exploración y son eficientes computacionales; y aunque no son exactos, es decir, no siempre hallan las soluciones óptimas, solo soluciones buenas, éstas soluciones son suficientes al momento de enfrentar un problema real.

En el capítulo 3 se presentó un tipo de problema de optimización, el Problema Inverso, el cual es muy utilizado en identificación de parámetros, y se expusieron diferentes métodos de solución, donde algunos exigen tener buen análisis de los datos si se utiliza una técnica tradicional, mientras que a través de la técnica expuesta sólo se requiere conocer el modelo matemático del problema y los datos observados, y con esta información es suficiente para poder estimar los parámetros; esto se pudo observar en el capítulo 4 donde se resolvió el modelo logístico para la población de la ciudad de Medellín y Antioquia, proporcionando mejores soluciones que la expuesta en [18], muy cercanas a los datos reales y a las estimaciones realizadas por el DANE para los años siguientes.

Problemas abiertos

Cuando se resuelve un problema de optimización a través de métodos computacionales, usualmente queda una pregunta por resolver: ¿Qué tan buena es la solución?. ¿De qué manera se podría proporcionar mayor información al decisor sobre el nivel de confiabilidad de la solución presentada?.

En el último capítulo se presentó la aplicación para el modelo logístico de la población de Medellín, aquí sólo se llegó hasta la estimación de los parámetros del modelo matemático, lo cuál permite realizar proyección de la población, pero, si se ligan estas proyecciones con condiciones como: estratos económicos y ubicación geográfica, entre otras, podría ser una buena fuente de información al momento de querer realizar por ejemplo, planeación en educación, salud y vivienda de una ciudad, o en el caso de una institución educativa, podría estimarse la demanda que tendrá para años futuros.

Bibliografía

- [1] ALEFELD, G. y Hersberger, J. Introduction to interval Computations, Academic Press, (1983).
- [2] ANGELINE, Peter, Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In V.W. Porto, N. Saravanan, D. Waagen, and A.E. Eiben, editors, Evolutionary Programming VII. 7th International Conference, EP 98, pp 601-610. Springer. Lecture Notes in Computer Science Vol. 1447, San Diego, California, USA (1998)
- [3] CHONG, Li, Wen-Hong Zhang y Xiao-Qing Jin, Convergence and uniqueness properties of Gauss-Newton's method, Computers and Mathematics with Applications, Vol. 47, ISSN 0898-1221, doi: 10.1016/S0898-1221(04)90086-7, pp 1057-1067 (2004)
- [4] COELLO, Carlos. Constraint-handling techniques used with evolutionary algorithms. In Proceedings of the 2007 GECCO Conference Companion on Genetic and Evolutionary Computation (London, United Kingdom, July 07 - 11, 2007). GECCO '07. ACM, New York, NY, 3057-3077. DOI= <http://doi.acm.org/10.1145/1274000.1274105> (2007)
- [5] COELLO C., Lamont G. y Van Veldhuizen D. Evolutionary Algorithms for Solving Multi-Objective Problems. Second Edition. Springer, (2007).
- [6] HADJ-ALOUANE, B. y Bean, J. A Genetic Algorithm for the Multiple-Choice Integer Program, Operation Research, vol 45, pp 92-101 (1997)
- [7] HERNÁNDEZ, A. Botello, S., Lizárraga, G. y Coello, C. IS-PAESs a constraint-handling technique based on multiobjective optimization concepts, EMO'03: Proceedings of the 2nd international conference on Evolutionary multi-criterion optimization, ISBN 3-540-01869-7, pp. 73-87 (2003)
- [8] HOMAIFAR, A., Qi, C. y Lai S. Constrained Optimization Via Genetic Algorithms. SIMULATION. Vol. 62, doi:10.1177/003754979406200405, pp 242-253 (1997)
- [9] KANZOW, Christian, Yamashita, Nobuo y Fukushima, Masao, Levenberg-Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints, Journal of Computational and Applied Mathematics, Vol. 172, ISSN 0377-0427, doi: 10.1016/j.cam.2004.02.013, pp 375-397 (2004).

- [10] ISHIBUCHI, H. y Tanaka, H. Multiobjective programming in optimization of the interval objective function, *European Journal of Operational Research* 48, pp 219-225 (1990)
- [11] JIANG, C., Han*, X., Guan, F.J. y Li, Y.H., An uncertain structural optimization method based on nonlinear interval number programming and interval analysis method, *Engineering Structures*, vol. 29, doi:10.1016/j.engstruct.2007.01.020, pp. 3168-3177 (2007)
- [12] JOINES, J.A., Houck, C.R., On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's, *IEEE World Congress on Computational Intelligence, Proceedings of the First IEEE Conference on Evolutionary Computation*, vol., no., pp.579-584 vol.2, doi: 10.1109/ICEC.1994.349995 (1994)
- [13] LEDESMA VILLALBA, Alberto, *Identificación de parámetros en geotecnia aplicación a la excavación de túneles*. 1987
- [14] LI, F. y Jin, C. Study on fuzzy optimization methods based on principal operation and inequity degree *Computer Mathematical Applications*. DOI=<http://dx.doi.org/10.1016/j.camwa.2008.02.042> (2008)
- [15] MERKLE L. y Lamont G. A Random Function Based Framework for Evolutionary Algorithms. *Proceedings of the Seventh International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers, San Mateo, California, pp. 105-112 (1997)
- [16] MOORE, Ramon. *Method and Applications of Interval Analysis*. SIAM (1979)
- [17] PAREDIS, J. *Co-evolutionary Constraint Satisfaction*. Springer (1994)
- [18] POVEDA, Gabriel y Manrique, Jorge. Aplicación de la curva logística a los censos de la ciudad de Medellín. *Ecos de Economía*, no. 25 (2007): 7- 60.
- [19] POWELL, D. y Skolnick, M. Using Genetic Algorithms in Engineering Design Optimization with Non-Linear Constraints, *Proceedings of the 5th International Conference on Genetic Algorithms*, ISBN 1-55860-299-2, pp. 424-431 (1994)
- [20] RÅDSTRÖM, Hans. An embedding theorem for spaces of convex sets, *American mathematical Society*, vol. 3, pp. 165-169 (1953).
- [21] REYES-SIERRA, Margarita y Coello, Carlos. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*. ISSN 0973-1873 Vol. 2, No. 3, pp. 287-308 (2006)
- [22] SAKAWA, M. *Genetic Algorithms and Fuzzy Multiobjective Optimization*. Kluwer Academic Publishers. (2001)
- [23] SEDIGHIZADEH D. y Masehian E. Particle Swarm Optimization Methods, Taxonomy and Applications. *International Journal of Computer Theory and Engineering*. pp 486-502 (2009)
- [24] SHIN, Y.S. y R.V. Grandhi. A global structural optimization technique using an interval method *Structural and Multidisciplinary Optimization*, Vol 22, Springer doi:10.1007/s00158-001-0155-1 (2001)

- [25] SURRY, P. y Radcliffe, N. The COMOGA Method: Constrained Optimisation by Multi-Objective Genetic Algorithms, *Control and Cybernetics*, vol. 26, pp 391-412 (1997)
- [26] TARANTOLA, A. *Inverse Problem Theory and Methods for Model Parameter Estimation*, SIAM (2005)
- [27] TARANTOLA, A. Popper, Bayes and the inverse problem. *Nature Physics*, vol. 2, pp. 492-494, doi:10.1038/nphys375 (2006)
- [28] VELDHUIZEN, D. y Lamont G. Multiobjective Evolutionary Algorithm Research: A History and Analysis, pp 1-88 (1998):
- [29] WOLPERT, David y Macready, William. No Free Lunch Theorems for Optimization, *IEEE Transaction on Evolutionary Computation*, Vol 1, pp 67-82 (1997)
- [30] WU, Hsien-Chung. The Karush-Kuhn-Tucker optimality conditions in an optimization problem with interval-valued objective function, *European Journal of Operational Research*, vol 176, pp 46-59 doi:10.1016/j.ejor.2005.09.007 (2007)
- [31] WU, Hsien-Chung. The Karush-Kuhn-Tucker optimality conditions in multi objective programming problems with interval-valued objective functions, *European Journal of Operational Research*, 196, pp. 49-60 doi:10.1016/j.jmaa.2007.05.023, (2009)
- [32] WU, Hsien-Chung- On interval-valued nonlinear programming problems, *Journal of Mathematical Analysis and Applications*, vol 338, pp 299-316 doi:10.1016/j.jmaa.2007.05.023, (2008)
- [33] ZITZLER E., Laumanns M. y Thiele L., SPEA2: Improving the Strength Pareto Evolutionary Algorithm. *Optimization*. pp 1-19 (2001).
- [34] ZITZLER E., Laumanns M. y Bleuler S. A Tutorial on Evolutionary Multi objective, In: Gandibleux X., (eds.): *Metaheuristics for Multiobjective Optimisation*. *Lecture Notes in Economics and Mathematical Systems*, Vol. 535, Springer (2004)