



**INSTITUTO POLITÉCNICO NACIONAL**

---

---

**ESCUELA SUPERIOR DE FÍSICA Y MATEMÁTICAS**

**Métodos matemáticos y  
heurísticos para optimización  
con dos funciones objetivo.**

**T E S I S**

**PARA OBTENER EL GRADO DE:**

**MAESTRO EN CIENCIAS  
FISICOMATEMÁTICAS**

**PRESENTA:**

**Uribe Richaud Lourdes Fabiola.**

**ASESOR: Dra. Lara López Adriana.**

**Cd. De México,**

**Mayo 2016.**



# Resumen.

La optimización multiobjetivo (OM) ataca problemas que surgen de manera natural en distintas áreas del conocimiento, donde se requiere satisfacer más de un criterio a la vez y estos suelen estar en conflicto entre sí. La solución de este tipo de problemas no es única, por lo que para la toma de decisiones es usual presentar al usuario más de una alternativa que optimice todos los criterios considerados de manera simultánea. Tiene sentido en este contexto el desarrollo de métodos numéricos para la aproximación del conjunto que contiene dichas soluciones. En esta tesis se presenta un estudio teórico y experimental de diversos métodos que destacan en la literatura de OM. Se propone al final una metodología híbrida que combina un algoritmo del tipo heurístico con un buscador local, mediante una técnica especializada para el archivado de las soluciones. En dicha propuesta se aprovecha información disponible de la función objetivo para acelerar la convergencia de la heurística hacia la solución buscada (sin perder la diversidad de las soluciones y con un bajo costo computacional). El desempeño de todos los métodos que se presentan a lo largo de la tesis se discute mediante ejemplos numéricos.



# Abstract.

Multi-objective optimization (MO) addresses problems that naturally arise in different areas where it is necessary to satisfy more than one criterion at time; usually these criteria are in conflict with each other. For decision makers it is convenient to get a set of “compromise solutions” since the solution of a MO problem is not unique. Nowadays, the design of numerical procedures to approximate multi-objective solution sets is an interesting subject. In this thesis, a theoretical and experimental study of several MO notable methods is presented. Besides, a new hybrid method that combines a heuristic algorithm with a local search method is proposed. This interleaving is possible made by the use of a specialized solution archiving strategy. This proposed method exploits available information of the test function in order to speed up the heuristic method convergence towards the solution—while maintaining solution spread and at a low computational cost. The performance of all the methods discussed throughout this work is explained by numerical examples.



*Para mis padres y abuelos.*  
Porque ustedes me ayudaron a ser lo que soy.



# Agradecimientos

Agradezco principalmente a mis padres y a mis hermanos, porque siempre han estado presentes, me han apoyado incondicionalmente en todas mis decisiones y me han dado ánimos durante todo este proceso. También quiero agradecer a mi familia, por todo el apoyo brindado con el paso de los años y porque sé que siempre estarán ahí.

A mi asesora, la Dra. Adriana Lara, por toda la confianza depositada en mí a lo largo de todos los años de conocernos y de trabajar juntas. Gracias por todo lo que me ha enseñado, no sólo académicamente sino humanamente. Agradezco infinitamente todas nuestras conversaciones, en las que puedo encontrar un consejo, una enseñanza o un momento de relajación. Por enseñarme lo bonito que es la investigación y por recordarme los motivos por los que me decidí por esta área. Agradezco también al Dr. Alin Carsteanu, al Dr. Egor Maximenko, al Dr. Juan Ignacio Rodríguez y al M. en C. Víctor Hugo Ibarra por leer esta tesis y por sus valiosos comentarios y observaciones.

A todos mis amigos, por todos los momentos vividos y sobre todo por dejarme ser parte de sus vidas. Un agradecimiento especial a Oliver Cuate, por su amistad y apoyo a lo largo de todos estos años. Gracias por toda la ayuda brindada y por hacerme parte de la familia Cinvestav. También quiero agradecer a mi mejor amigo, Omar Osorio, por creer en mí y estar conmigo siempre. Por todas las aventuras, momentos y risas. Y por enseñarme lo que es tener un amigo incondicional. A Fernando Martínez y Edgar Sandoval, gracias por todo. A mis colegas de la maestría: Juan Esaú Trejo, Marco Amaral y Eduardo Merín por ser un gran equipo de estudio y de debate.

Agradezco al CONACyT por la beca económica otorgada para la realización de estos estudios. Esta tesis se derivó de los proyectos de investigación SIP20140664, SIP20152082 y de la Beca de Estímulo Institucional de Formación de Investigadores BEIFI del IPN.



# Índice

<b>Lista de Figuras</b>	<b>XI</b>
<b>Lista de Tablas</b>	<b>XV</b>
<b>Lista de Acrónimos</b>	<b>XIX</b>
<b>1 Preliminares.</b>	<b>5</b>
1.1 El Problema de Optimización Multiobjetivo. . . . .	5
1.2 Vectores compromiso . . . . .	6
1.3 Condiciones de Optimalidad . . . . .	8
1.4 Indicadores de Desempeño. . . . .	11
<b>2 Métodos agregativos</b>	<b>15</b>
2.1 Método de suma de pesos . . . . .	15
2.1.1 Descripción . . . . .	15
2.1.2 Discusión . . . . .	18
2.1.3 Resultados numéricos . . . . .	22
2.2 Método de Suma de Pesos Adaptativo . . . . .	23
2.2.1 Preliminares . . . . .	23
2.2.2 Descripción . . . . .	24
2.2.3 Discusión . . . . .	27

2.2.4	Experimentos numéricos . . . . .	31
<b>3</b>	<b>Métodos de punto de referencia y continuación</b>	<b>33</b>
3.1	Método NBI . . . . .	33
3.1.1	Preliminares . . . . .	33
3.1.2	Descripción . . . . .	34
3.1.3	Discusión . . . . .	36
3.1.4	Experimentos numéricos . . . . .	43
3.2	Método Pareto Tracer . . . . .	45
3.2.1	Preeliminaries . . . . .	45
3.2.2	Descripción . . . . .	48
3.2.3	Discusión . . . . .	50
3.2.4	Experimentos numéricos . . . . .	53
<b>4</b>	<b>Método poblacional</b>	<b>57</b>
4.1	Introducción . . . . .	57
4.2	Preliminares . . . . .	58
4.3	Algoritmo NSGA-II . . . . .	60
4.4	Resultados numéricos . . . . .	69
<b>5</b>	<b>Método Híbrido.</b>	<b>71</b>
5.1	Análisis preliminar . . . . .	71
5.2	Descripción del método . . . . .	72
5.2.1	Módulo Archivador . . . . .	73
5.2.2	NSGA-II + PT. . . . .	73
5.3	Discusión . . . . .	74
5.4	Análisis numérico . . . . .	79
5.5	Resultados . . . . .	80

<i>ÍNDICE</i>	XI
<b>6 Conclusiones y Trabajo Futuro.</b>	<b>85</b>
<b>Bibliografía</b>	<b>91</b>



# Índice de figuras

1.1	Representación de funciones en conflicto. . . . .	6
1.2	Ejemplo de dominancia de Pareto. . . . .	7
1.3	Ejemplo de óptimos débiles de Pareto. . . . .	9
2.1	Resultados experimentales de la función Binh usando SP . . . . .	17
2.2	Resultados experimentales de la función Fonseca usando SP . . . . .	18
2.3	Interpretación Geométrica de SP. . . . .	20
2.4	Resultados experimentales de la función Poloni usando SP . . . . .	21
2.5	Resultados experimentales de la función Rendon 2 usando SP . . . . .	22
2.6	Representación del caso donde el FP es convexo con curvaturas no uniformes. . . . .	24
2.7	Representación del caso donde la frontera es no convexa con soluciones no dominadas. . . . .	25
2.8	Representación del caso donde el FP es no convexo de soluciones dominadas. . . . .	25
2.9	Resultados experimentales de la función Fonseca usando SPA . . . . .	27
2.10	Resultados experimentales de la función Lis usando SPA . . . . .	28
2.11	Resultados experimentales de la función Poloni usando SPA . . . . .	29
2.12	Resultados experimentales de la función Rendon 2 usando SPA . . . . .	29
2.13	Resultados experimentales de la función Schaffer 2 usando SPA . . . . .	30

3.1	Esquema del NBI . . . . .	35
3.2	Resultados experimentales de la función Binh usando NBI . . . .	39
3.3	Resultados experimentales de la función Fonseca usando NBI . .	40
3.4	Resultados experimentales de la función de Poloni usando NBI .	40
3.5	Resultados experimentales de la función Schaffer 2 usando NBI .	41
3.6	Resultado experimental de la función ZDT1 usando NBI para $x_1^{*1}$	43
3.7	Resultado experimental de la función ZDT1 usando NBI para $x_3^{*1}$	44
3.8	Resultado experimental de la función ZDT1 usando NBI para $x_3^{*1}$	44
3.9	Resultados experimentales de la función de Dent usando PT . . .	51
3.10	Resultados experimentales de la función de Fonseca usando PT .	51
3.11	Resultados experimentales de la función de Murata usando PT .	52
3.12	Resultado experimental de la función ZDT1 usando PT . . . . .	53
3.13	Resultado experimental de la función ZDT3 usando PT . . . . .	54
4.1	Ejemplo de la jerarquización de Pareto . . . . .	63
4.2	Resultados experimentales de la función Dent usando NSGA-II con 600 generaciones . . . . .	64
4.3	Resultados experimentales de la función Dent usando NSGA-II con 1200 generaciones . . . . .	65
4.4	Resultados experimentales de la función Lis usando NSGA-II con 600 generaciones . . . . .	65
4.5	Resultados experimentales de la función Lis usando NSGA-II con 1200 generaciones . . . . .	66
4.6	Resultados experimentales de la función Poloni usando NSGA-II con 600 generaciones . . . . .	67
4.7	Resultados experimentales de la función Poloni usando NSGA-II con 1200 generaciones . . . . .	67
4.8	Resultados experimentales de la función Schaffer 2 usando NSGA-II con 600 generaciones . . . . .	68
4.9	Resultados experimentales de la función Schaffer 2 usando NSGA-II con 1200 generaciones . . . . .	68

4.10	Resultado experimental de la función ZDT3 usando NSGA-II con 600 y 1200 generaciones . . . . .	69
5.1	Resultados experimentales de la función Dent usando NSGA-II + PT . . . . .	76
5.2	Resultado experimental de la función ZDT1 usando NSGA-II + PT . . . . .	77
5.3	Resultado experimental de la función ZDT3 usando NSGA-II + PT . . . . .	77
5.4	Resultado experimental de la función ZDT4 usando NSGA-II + PT . . . . .	78
5.5	Resultado experimental de la función Laumanns . . . . .	83
5.6	Resultado experimental de la función ZDT1 . . . . .	83
5.7	Resultado experimental de la función ZDT2 . . . . .	84
5.8	Resultado experimental de la función ZDT3 . . . . .	84



# Índice de tablas

2.1	Resultados numéricos del Método Suma de Pesos. . . . .	23
2.2	Resultados numéricos del Método Suma de Pesos Adaptativo. . .	32
3.1	Resultados numéricos del Método NBI. . . . .	45
3.2	Resultados numéricos del Método Pareto Tracer. . . . .	55
4.1	Resultados numéricos del NSGA-II. . . . .	70
5.1	Tabla de valores para los parámetros $\varepsilon$ . . . . .	80
5.2	Costo computacional. . . . .	80
5.3	Resultados para $\Delta_2$ . . . . .	82



# Acrónimos

**POM:** *Problema de Optimización Multiobjetivo*

**POE:** *Problema de Optimización Escalar*

**CP:** *Conjunto de Pareto*

**FP:** *Frente de Pareto*

**SP:** *Suma de Pesos*

**SPA:** *Suma de Pesos Adaptativo*

**NBI:** *Normal Boundary Intersection*

**PT:** *Pareto Tracer*

**PC:** *Predictor-Corrector*

**KKT:** *Karush-Kuhn-Tucker*

**AG:** *Algoritmo Genético*

**AGs:** *Algoritmos Genéticos*

**AGMO:** *Algoritmo Genético Multiobjetivo*

**AGMOs:** *Algoritmos Genéticos Multiobjetivo*

**NSGA-II:** *Non-dominating Sorting Genetic Algorithm II*

**MAG:** *Micro Algoritmo Genético*

**PAES:** *Pareto Archived Evolution Strategy*

**DG:** *Distancia Generacional*

**DGI:** *Distancia Generacional Invertida*



# Introducción.

Los problemas descritos mediante la optimización multiobjetivo surgen de manera natural en diversas áreas del conocimiento; se pueden encontrar ejemplos de estos en finanzas, economía, ingeniería, entre otras. En general, la optimización tiene aplicación en todas las áreas donde se busca tener un mayor beneficio a través de los recursos con los que se cuenta; debido a que usualmente estos recursos son limitados. En la optimización multiobjetivo se busca sacar el mejor provecho (optimizar) considerando más de un criterio a la vez; dichos criterios suelen estar en conflicto entre sí. Un ejemplo tradicional de esta situación es el problema de optimización de portafolios, donde se busca minimizar el riesgo de cierta inversión a la vez que se obtiene el mayor rendimiento. Tal tipo de problema se denomina *Problema de Optimización Multiobjetivo* (POM).

A la par del acelerado desarrollo del cómputo actual, la optimización numérica ha evolucionado en algoritmos cada vez más complejos llevando así a soluciones aceptables de ciertos problemas que parecerían intratables hace algunos años. En la literatura actual existen diversos métodos para resolver POMs. Cada método busca dar solución a problemas cada vez más generales, (*i.e.* con planteamientos más complejos, con menos suposiciones respecto a las funciones objetivo, etc.) de la manera más satisfactoria posible. Una característica básica de este tipo de problemas, es que no se busca aproximar una única solución al POM, sino calcular todo un conjunto de *soluciones compromiso* entre los criterios a optimizar. Es decir, encontrar aquellos puntos de equilibrio en los que no es posible mejorar en alguno de los criterios sin empeorar el desempeño de la solución respecto a los demás. Los métodos que exponemos en esta investigación tienen como meta aproximar numéricamente dicho conjunto de soluciones compromiso (puntos de equilibrio) de manera numéricamente eficiente.

En este documento se presenta un estudio teórico de algunos métodos, para resolver POMs, disponibles en la literatura. Se hace un estudio experimental para determinar su desempeño, en problemas con características diversas, y determinar posibles complicaciones que surgen en su implementación práctica. Se proponen dos posibles métodos híbridos que resuelvan el POM de manera eficiente.

### Objetivo

El objetivo general de esta tesis es realizar un estudio teórico y experimental de algunos métodos para la solución numérica de POMs, disponibles en la literatura reciente. Nos limitaremos a considerar dos funciones objetivo continuas a optimizar de manera simultánea, sin restricciones activas. Como objetivos particulares tenemos:

1. Hacer un estudio de los métodos actuales para resolver el POM. Exponer las ventajas, condiciones y desventajas de cada método estudiado, así como determinar los aspectos prácticos para su correcta implementación computacional.
2. Explorar algunas posibles hibridaciones de métodos poblacionales (*i.e.* heurísticas de computación evolutiva) con métodos exactos (*i.e.* métodos de Programación Matemática tradicional).

### Metodología.

La metodología seguida en este trabajo fue:

- Llevar a cabo un estudio teórico de los métodos tanto exactos como heurísticos, disponibles en la literatura, para la solución de POMs.
- Llevar a cabo la implementación computacional de todos los métodos estudiados, para explorar sus debilidades, y complicaciones en la práctica.
- Llevar a cabo un estudio empírico de la eficacia y eficiencia de dichos métodos sobre problemas de prueba estándar.
- Proponer la construcción de un método híbrido que subsane algunas deficiencias de los métodos estudiados.
- Finalmente, explorar la efectividad y eficiencia de la propuesta de método híbrido, sobre problemas de prueba estándar.

### Contribuciones

Este trabajo contribuye a la generación de conocimiento en el área de optimización multiobjetivo, con énfasis en el caso particular de dos funciones objetivo (biobjetivo). Las contribuciones particulares se listan a continuación:

1. Se presenta un análisis teórico y experimental detallado de los métodos: *Suma de Pesos* (SP), *Suma de Pesos Adaptativo* (SPA), *Normal Boundary Intersection* (NBI), *Pareto Tracer* (PT) y *Non-dominating Sorting Genetic Algorithm II* (NSGA-II).
2. Se proponen dos tipos particulares de métodos de archivado de soluciones para métodos poblacionales (*i.e. heurísticos*), que hacen uso del método PT, previamente estudiado.
3. Se propone un método híbrido original, que conjunta el NSGA-II, con los métodos de archivado propuestos.
4. Se incluye, adjunto a esta tesis, el software desarrollado con la realización de todos los métodos aquí estudiados: los exactos, los poblacionales y los híbridos propuestos.

## Organización del documento.

Esta tesis está conformada por 7 capítulos, incluyendo este capítulo de introducción. En el capítulo 2, se muestran algunos conceptos básicos de la optimización multiobjetivo que serán útiles a lo largo del trabajo. En el capítulo 3 y 4 se presentan los métodos exactos estudiados; para cada método se incluye un análisis de propiedades, ventajas y desventajas; además de que se incluyen ejemplos numéricos para mostrar su comportamiento sobre distintas funciones de prueba. En el capítulo 5 se presenta un algoritmo evolutivo multiobjetivo, como método poblacional heurístico. En el capítulo 6 se presenta el método híbrido propuesto, haciendo énfasis en su funcionamiento; así como los resultados de los experimentos numéricos y las debidas comparaciones. Finalmente, las conclusiones de esta tesis y las posibles líneas de investigación a seguir se explican en el capítulo 7.

#### 4 ÍNDICE DE TABLAS

---

# Capítulo 1

## Preliminares.

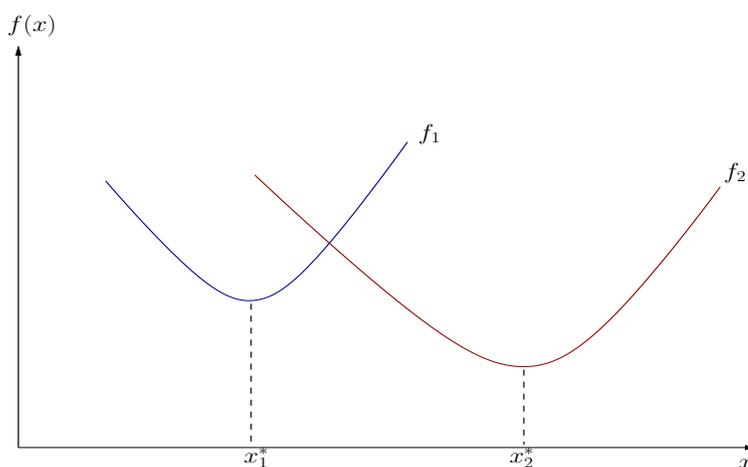
### 1.1 El Problema de Optimización Multiobjetivo.

El *problema de Optimización Multiobjetivo* (POM) busca optimizar (minimizar o maximizar) de manera simultánea  $m$  funciones ( $m \geq 2$ ), sobre un mismo dominio. En este trabajo consideramos únicamente problemas *sin restricciones*, por lo que nos referimos como POM al siguiente problema concreto:

$$\text{Minimizar } F : \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad (1.1)$$

donde las funciones  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$  se conocen como las *funciones objetivo* y se denomina *vector objetivo* a  $[f_1(x), f_2(x), \dots, f_m(x)]^T$ . El espacio  $\mathbb{R}^m$ , contradominio de la función  $F$ , se referirá como el *espacio de los objetivos*. Supondremos a lo largo de este trabajo, sin pérdida de generalidad, que todas las funciones objetivo se minimizan; dado que maximizar  $f$  equivale a minimizar  $-f$ .

La principal diferencia entre resolver un problema de optimización tradicional (del tipo *Minimizar*  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ), al cual nos referiremos aquí como *Problema de Optimización Escalar* (POE) y resolver un POM radica en la naturaleza de su solución. En el POE se busca calcular un cierto vector de parámetros  $x^* \in \mathbb{R}^n$  correspondientes al valor óptimo de  $f$ , mientras que resolver un POM equivale a calcular todo un conjunto  $S \subset \mathbb{R}^n$  de *vectores compromiso* entre los valores de todas las funciones objetivo. Esto surge dada la suposición de que las funciones del vector objetivo se encuentran *en conflicto* entre sí, observe esto en la figura 1.1; por esta razón, resolver un POM se refiere justamente a encontrar dicho conjunto de soluciones.



**Figura 1.1:** En esta figura se muestra la gráfica de dos funciones que se encuentran en conflicto. Observe que al variar el parámetro  $x$  mientras  $f_1$  aumenta su valor,  $f_2$  lo disminuye. El conjunto de soluciones compromiso para este POM sería la línea que une  $x_1^*$  y  $x_2^*$ .

## 1.2 Vectores compromiso

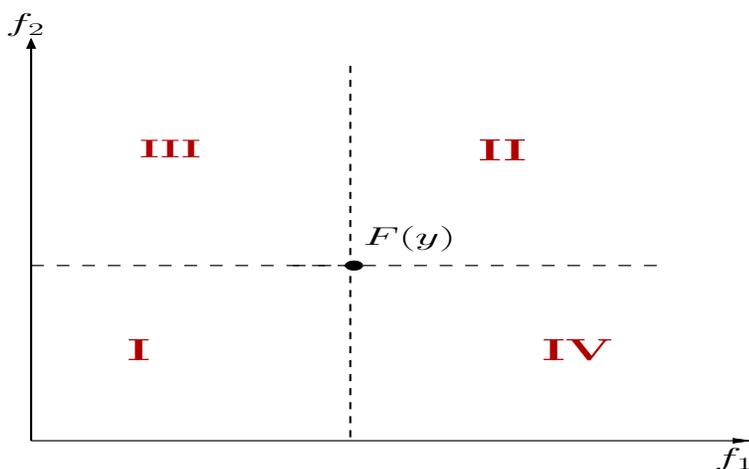
Para la búsqueda de las mejores soluciones a un POM es necesario definir una *relación de orden* entre los distintos vectores objetivo a comparar. En  $\mathbb{R}^m$  (con  $2 \leq m$ ) no se conoce una relación de orden total que considere a cada una de las funciones objetivo con la misma importancia<sup>1</sup>. En optimización multiobjetivo se considera entonces el concepto de *Dominancia de Pareto* para definir una relación de orden parcial entre las soluciones. Consideremos entonces la siguiente relación usual en  $\mathbb{R}^m$  tal que para todo  $x, y \in \mathbb{R}^m$  se tiene que  $x \leq y$  sí y sólo si  $x_i \leq y_i$  para todo  $i \in \{1, \dots, m\}$ ; es decir, que mejora o iguala su valor componente a componente. Lo relevante al tener un orden parcial, en vez de total, en el espacio de trabajo radica en que en algunos casos no es posible decidir que una cierta solución es mejor que otra. La relación de comparación de vectores que acabamos de mencionar es la base para la definición de Dominancia de Pareto que se enuncia a continuación:

**Definición 1.1.** Sean  $x, y \in \mathbb{R}^n$ , decimos que  $x$  domina a  $y$ , en el sentido de Pareto, si y sólo si  $F(x)$  es parcialmente menor a  $F(y)$ , ( $F(x) < F(y)$ ), como en (1.1). Es decir, que para toda componente  $i \in \{1, \dots, m\}$  se cumple que

$$f_i(x) \leq f_i(y) \text{ y } f_j(x) < f_j(y) \text{ para algún } j, \text{ con } j \in \{1, \dots, m\}.$$

Si esto pasa, se establece la notación  $x \preceq y$ .

<sup>1</sup>El orden usual lexicográfico induce una preferencia no deseada entre las funciones objetivo a comparar, por lo que está fuera de consideración en este análisis donde suponemos todas las funciones con la misma importancia para el usuario.



**Figura 1.2:** Esta figura muestra un ejemplo de dominancia de Pareto. La región I representa todos los puntos  $x$  que dominan a  $y$ . La región II representa todos los puntos  $x$  que son dominados por  $y$ . Las regiones III y IV representan todos los puntos  $x$  que son incomparables con  $y$ .

La Figura 1.2 ejemplifica la dominancia de Pareto. Si se tiene  $x \in \mathbb{R}^n$  tal que  $F(x) \in I$ , entonces  $x \preceq y$ . Si  $F(x) \in II$ , entonces  $y \preceq x$ . Finalmente, si  $F(x) \in III$  o  $F(x) \in IV$  entonces  $x$  y  $y$  son mutuamente no dominados, es decir que son *incomparables*.

**Proposición 1.1.** *La relación de Dominancia de Pareto induce un orden parcial estricto en  $\mathbb{R}^n$ .*

*Demostración.* Para demostrar que la dominancia de Pareto induce un orden parcial estricto se debe cumplir que la relación es: irreflexiva, transitiva y asimétrica.

- i) Sea  $x \in \mathbb{R}^n$ . Supongamos que es reflexiva,  $x \preceq x$ . Entonces  $f_i(x) \leq f_i(x)$  para  $i = 1, \dots, m$  y  $f_j(x) < f_j(x)$  para algún  $j \in \{1, \dots, m\}$  lo cual es una contradicción. Luego  $x \not\preceq x$ , por lo tanto la dominancia de Pareto es irreflexiva.
- ii) Sean  $x, y, z \in \mathbb{R}^n$ ,  $x \preceq y$  y  $y \preceq z$ . Entonces  $f_i(x) \leq f_i(y)$  para  $i = 1, \dots, m$  y  $f_j(x) < f_j(y)$  para algún  $j \in \{1, \dots, m\}$  y  $f_i(y) \leq f_i(z)$  para  $i = 1, \dots, m$  y  $f_j(y) < f_j(z)$  para algún  $j \in \{1, \dots, m\}$ . Por transitividad en  $\mathbb{R}$  se cumple que si  $f_i(x) \leq f_i(y)$  y  $f_i(y) \leq f_i(z)$  entonces  $f_i(x) \leq f_i(z)$  para  $i = 1, \dots, m$ . Análogamente  $f_j(x) < f_j(z)$  para algún  $j \in \{1, \dots, m\}$ , luego  $x \preceq z$ . Por lo tanto la dominancia de Pareto es transitiva.
- iii) Sean  $x, y \in \mathbb{R}^n$ . Supongamos que es simétrica,  $x \preceq y$  y  $y \preceq x$ . Entonces  $f_i(x) \leq f_i(y)$  para  $i = 1, \dots, m$  y  $f_j(x) < f_j(y)$  para algún  $j \in \{1, \dots, m\}$

y  $f_i(y) \leq f_i(x)$  para  $i = 1, \dots, m$  y  $f_j(y) < f_j(x)$  para algún  $j \in \{1, \dots, m\}$  lo cual es una contradicción. Por lo tanto la dominancia de Pareto es asimétrica.

Por *I*, *II*, *III* concluimos que la relación de Dominancia de Pareto induce un orden parcial estricto.  $\square$

De esta manera, la Dominancia de Pareto nos permite discernir entre soluciones del espacio de búsqueda (el dominio) del POM. A continuación hablaremos sobre las condiciones de optimalidad para este tipo de problemas.

### 1.3 Condiciones de Optimalidad

**Definición 1.2.** *Un punto  $x^* \in \mathbb{R}^n$  es un óptimo débil de Pareto si no existe  $x \in \mathbb{R}^n$  tal que  $F(x) < F(x^*)$ , es decir, no existe  $x$  tal que  $f_i(x) < f_i(x^*)$  para  $i \in \{1, \dots, m\}$ .*

**Definición 1.3.** *Un punto  $x^* \in \mathbb{R}^n$  se dice óptimo de Pareto si no existe  $x \in \mathbb{R}^n$  tal que  $x \preceq x^*$ .*

De la Figura 1.3 se aprecia la diferencia entre los óptimos débiles de Pareto y los óptimos de Pareto.

**Definición 1.4.** *El conjunto de todos los óptimos de Pareto se conocen como Conjunto de Pareto (CP).*

**Definición 1.5.** *La imagen de los puntos en el Conjunto de Pareto es llamado comúnmente Frente de Pareto (FP).*

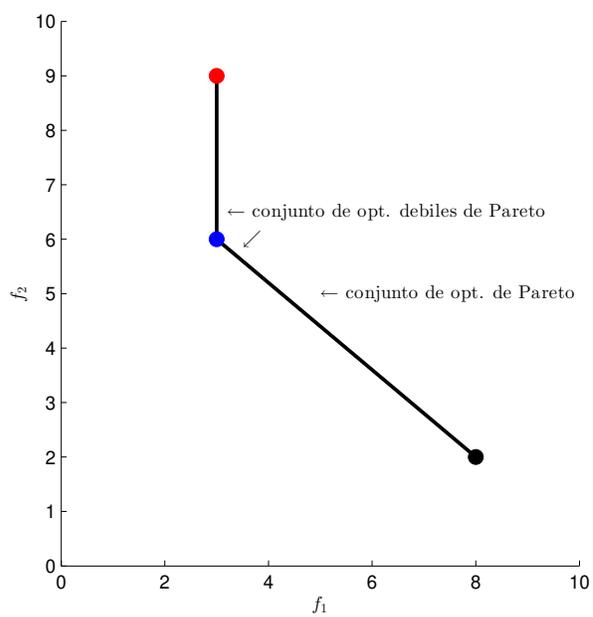
**Definición 1.6.** *Sea  $\Theta \subset \mathbb{R}^n$  un conjunto convexo si y solo si se cumple que para todos elementos  $u, v \in \Theta$ , el segmento de línea entre  $u$  y  $v$  está completamente contenido en  $\Theta$ . Es decir, que  $\alpha u + (1 - \alpha)v \in \Theta \quad \forall u, v \in \Theta$  con  $\alpha \in (0, 1)$ .*

**Definición 1.7.** *Una función  $f$  se dice convexa si su dominio es un conjunto convexo y si para cualesquiera dos puntos  $x$  y  $y$  en el dominio, la gráfica de  $f$  se localiza debajo de la línea que conecta  $(x, f(x))$  y  $(y, f(y))$  en el espacio  $\mathbb{R}^{n+1}$ , es decir que*

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) \quad \forall \alpha \in [0, 1].$$

**Definición 1.8.** *El problema de optimización multiobjetivo se dice ser **convexo** si todas las funciones objetivo y regiones factibles (en este caso el dominio) son convexas [Miettinen, 1999].*

Análogamente a la optimización escalar, en la optimización multiobjetivo existen condiciones de optimalidad. En la siguiente sección se habla sobre ellas.



**Figura 1.3:** Esta figura muestra un ejemplo de puntos débiles de Pareto y óptimos de Pareto.

**Teorema 1.1. Condición Suficiente de Karush-Kuhn-Tucker (KKT) para el POM sin restricciones.** Sea la función objetivo convexa y continuamente diferenciable en  $x^*$ . Una condición suficiente para que  $x^*$  sea un óptimo de Pareto es que existan multiplicadores  $\lambda > 0 \in \mathbb{R}^m$  tal que

$$\sum_{i=1}^m \lambda_i \nabla f_i(x^*) = 0. \quad (1.2)$$

Los puntos que cumplan dichas condiciones se llaman puntos de KKT.

*Demostración.* Sea  $\lambda > 0 \in \mathbb{R}^m$ . Definimos una función  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  como  $F(x) = \sum_{i=1}^m \lambda_i f_i(x)$  donde  $x \in S \subset \mathbb{R}^n$ .

Observe que  $F$  es convexa porque  $f_i(x)$  es convexa para todo  $i = 1, \dots, m$  por hipótesis, además  $\lambda > 0$ . Por (1.2) tenemos  $\nabla F(x^*) = 0$ . De acuerdo a la condición de optimalidad para optimización mono-objetivo si  $\nabla F(x^*) = 0$  entonces  $x^*$  es un punto mínimo de la función. De este modo

$$F(x^*) \leq F(x) \text{ para todo } x \in S, \quad (1.3)$$

$$\Rightarrow \sum_{i=1}^m \lambda_i f_i(x^*) \leq \sum_{i=1}^m \lambda_i f_i(x) \text{ para todo } x \in S. \quad (1.4)$$

Supongamos que  $x^*$  no es óptimo de Pareto. Entonces existe un  $\hat{x} \in S$  tal que  $f_i(\hat{x}) \leq f_i(x^*)$  para todo  $i = 1, \dots, m$  y  $f_j(\hat{x}) < f_j(x^*)$  para algún  $j \in \{1, \dots, m\}$ . Como  $\lambda_i > 0$  para todo  $i = 1, \dots, m$  tenemos

$$\sum_{i=1}^m \lambda_i f_i(\hat{x}) < \sum_{i=1}^m \lambda_i f_i(x^*),$$

lo cual es una contradicción a (1.4). Por lo tanto  $x^*$  es un óptimo de Pareto. □

**Teorema 1.2. Condiciones Suficientes de Segundo Orden.** Sean todos los objetivos del POM doblemente diferenciables en  $x^* \in \mathbb{R}^n$ . Una condición suficiente para que  $x^*$  sea óptimo de Pareto es que existe un vector  $\lambda \in \mathbb{R}^m$  tal que

$$\sum_{i=1}^m \lambda_i \nabla f_i(x^*) = 0.$$

$$\sum_{i=1}^m \lambda_i = 1.$$

$$\lambda_i \geq 0 \quad i = 1, \dots, m.$$

$$d^T \left( \sum_{i=1}^m \lambda_i \nabla^2 f_i(x^*) \right) d \leq 0 \quad i = 1, \dots, k.$$

Con  $d \in \mathbb{R}^n \setminus \{0\}$  tal que  $\nabla f_i(x^*)^T d \leq 0 \quad i = 1, \dots, k$ .

*Demostración.* Ver [Wang, 1991]. □

Si la función objetivo no es convexa, pero se mantiene (1.2) para  $\lambda > 0$  entonces se cumple la *condición necesaria para la optimalidad de Pareto de Fritz John* [Miettinen, 1999]. Para terminar, introducimos algunos conceptos básicos que se utilizarán a lo largo de este trabajo:

## 1.4 Indicadores de Desempeño.

Existe una gran cantidad de indicadores de desempeño para comparar las soluciones arrojadas por diversos métodos, cada uno de ellos se enfoca en tratar de describir de forma representativa algún aspecto de interés. Por ejemplo, proximidad al óptimo, distribución de las soluciones, entre otras. Para cierto problema realizando diversas operaciones entre los elementos de uno o dos frentes.

Se suelen tomar en cuenta las siguientes consideraciones cuando se diseña un buen indicador de desempeño para POM.

1. Medir la distancia del FP generado por el algoritmo empleado con respecto al verdadero FP. Esto suponiendo que se conoce dicho frente.
2. Medir la distribución de las soluciones encontradas, de forma que se pueda tener una distribución de soluciones no dominadas uniforme.

En este trabajo, se utilizan indicadores de desempeño que conocen el FP real de las funciones de prueba. Entonces este tipo de indicadores, comparan el FP obtenido con el FP real y a partir de ciertas medidas de error indican la efectividad del algoritmo analizado. Los indicadores seleccionados son: Distancia Generacional, Distancia Generacional Invertida y  $\Delta_p$ , los cuales se describen a continuación.

## Distancia Generacional.

La *Distancia Generacional*. (DG) es un indicador de desempeño introducido por Van Veldhuizen & Lamont[Van Veldhuizen and Lamont, 1998] en 1998 como una manera de estimar qué tan lejos están los elementos del FP aproximado del FP teórico. Se define como:

$$DG := \frac{(\sum_{i=1}^n d_i^p)^{\frac{1}{p}}}{|FP_{aproximado}|},$$

donde  $|FP_{aproximado}|$  es el número de vectores en el FP aproximado y  $d_i$  es la distancia entre el  $i$ -ésimo miembro del FP aproximado y el miembro del FP teórico más cercano a él. Un valor de  $DG = 0$  indica que todos los elementos obtenidos están el FP real. Luego cualquier otro valor indica que tan lejos se encuentra la solución obtenida del FP real. Este indicador mide convergencia al FP real, no dispersión.

Una desventaja notoria de la DG es su forma de calcular la proximidad al FP, esta medida no penaliza la cantidad de puntos que existen en el FP actual ni su dispersión, si se encuentran cerca, o sobre el FP real. De este modo, un método que genere un único punto sobre el FP tendrá un mejor valor que otro que genere más puntos que estén cerca de la curva.

## Distancia Generacional Invertida.

La *Distancia Generacional Invertida*. (DGI) es un indicador propuesto por Cruz y Coello[Coello and Cortés, 2005]. Este indicador calcula una distancia promedio entre el FP aproximado y el FP teórico. Se define como:

$$DGI := \frac{(\sum_{i=1}^m d_i^p)^{\frac{1}{p}}}{|FP_{teorico}|},$$

donde  $|FP_{teorico}|$  es el número de vectores en el FP teórico y  $d_i$  es la distancia entre el  $i$ -ésimo miembro del FP teórico y el miembro del FP aproximado más cercano a él. Se puede considerar un complemento de la DG.

Este indicador corrige la desventaja que tiene la DG cuando un método devuelve un solo punto sobre el FP real. La DGI tendrá un valor grande en este caso puesto que todas las distancias que se promediarán serán las que hay entre dicho punto y cada uno de los elementos del FP real. Conforme más distantes estén estos elementos del punto obtenido, estas distancias crecerán, por ende el valor del indicador. Una desventaja que presenta este indicador es que no penaliza a las soluciones del FP actual que estén lejos del FP real siempre y cuando las demás soluciones estén cerca del FP real y cuenten con una buena dispersión, a diferencia de la DG.

### Distancia de Hausdorff: Indicador $\Delta_p$ .

En [Schütze et al., 2012] se propuso  $\Delta_p$  como un indicador de desempeño que mide la distancia promedio de Hausdorff entre dos conjuntos. Mide la distancia entre el conjunto formado por el FP aproximado  $A$  y el FP teórico  $B$ . Para esto utiliza la  $DG_p$  y la  $DGI_p$  definidas por:

$$DG_p(A, B) := \frac{1}{N} \left( \sum_{i=1}^N \text{dist}(a_i, B)^p \right)^{1/p},$$
$$DGI_p(A, B) := \frac{1}{M} \left( \sum_{i=1}^M \text{dist}(b_i, A)^p \right)^{1/p},$$

y que dan lugar a:

$$\Delta_p := \max(DG_p(A, B), DGI_p(A, B)).$$

Para fines de este trabajo, utilizamos  $p = 2$ , de manera que se aplica la distancia euclidiana entre dos puntos. Cuando se quiere aproximar el CP se tienen dos objetivos:

- Minimizar la distancia entre la aproximación final y el FP verdadero del problema (a veces se le llama *teórico*).
- Generar una muestra diversa de la aproximación final, que cubra todo el FP teórico y que cuente con una buena distribución.

Estas serían las dos metas que se desean cumplir con cada uno de los métodos que se estudian en los capítulos posteriores.



## Capítulo 2

# Métodos agregativos

### 2.1 Método de suma de pesos

El método conocido como *Suma de Pesos* (SP) es la alternativa más comúnmente utilizada para solucionar el POM. La idea detrás de este método es transformar el POM en un POE usando una ponderación de las funciones objetivo.

#### 2.1.1 Descripción

El método SP busca minimizar una combinación convexa de las diferentes funciones objetivo que forman el vector objetivo. En otras palabras, transforma el problema multiobjetivo en uno de optimización tradicional mediante una función de agregación  $S_F : \mathbb{R}^n \rightarrow \mathbb{R}$  de las distintas funciones objetivo, el POM transformado queda descrito por:

$$\begin{aligned} \text{Minimizar} \quad & S_F(x) := \sum_{i=1}^m w_i f_i(x) \\ \text{sujeto a} \quad & \sum_{i=1}^m w_i = 1 \\ & 0 \leq w_i \quad i \in \{1, \dots, m\}. \end{aligned} \tag{2.1}$$

Para encontrar la solución del problema (2.1) puede usarse cualquier método conveniente de optimización tradicional. Cada vector de pesos  $w$  determinado define un cierto POE particular. En principio, se podría pensar que entre más ponderaciones distintas sean consideradas para resolver el problema, más puntos del CP se tendrán; sin embargo esto no es así, como veremos posteriormente.

Los teoremas siguientes analizan la obtención de óptimos de Pareto mediante este método.

**Teorema 2.1.** *La solución del problema de suma de pesos (2.1) corresponde a un óptimo débil de Pareto del POM original.*

*Demostración.* [Miettinen, 1999] Sea  $x^* \in \mathbb{R}^n$  una posible solución al problema de suma de pesos. Procediendo por contradicción, supongamos que  $x^*$  no es un óptimo débil de Pareto. Esto quiere decir que existe  $x \in \mathbb{R}^n$  tal que  $f_i(x) < f_i(x^*)$  para todo  $i \in \{1, \dots, m\}$ . Dado que la definición del problema indica  $0 \leq w_i$  para todo  $i \in \{1, \dots, m\}$  y  $\sum_{i=1}^m w_i = 1$ ; se tiene que  $0 < w_j$  para al menos un  $j \in \{1, \dots, m\}$ , lo que implica que  $w_j f_j(x) < w_j f_j(x^*)$ . En consecuencia se cumple lo siguiente

$$\sum_{i=1}^m w_i f_i(x) < \sum_{i=1}^m w_i f_i(x^*),$$

lo cual es una contradicción al supuesto que  $x^*$  es solución del problema. Por lo tanto  $x^*$  debe ser un óptimo débil de Pareto.  $\square$

**Teorema 2.2.** *La solución del problema de suma de pesos (2.1) corresponde a un óptimo de Pareto si las componentes del vector de pesos son coeficientes positivos, es decir,  $0 < w_i$  para todo  $i \in \{1, \dots, m\}$ .*

*Demostración.* [Miettinen, 1999] Sea  $x^* \in \mathbb{R}^n$  una solución al problema de suma de pesos con coeficientes positivos en el vector de pesos. Suponga que  $x^*$  no es un óptimo de Pareto. Esto quiere decir que existe cierto  $x \in \mathbb{R}^n$  tal que  $f_i(x) \leq f_i(x^*)$  para todo  $i \in \{1, \dots, m\}$  y que  $f_j(x) < f_j(x^*)$  para al menos un  $j \in \{1, \dots, m\}$ . Dado que  $w_i > 0$  para todo  $i \in \{1, \dots, m\}$ , tenemos que

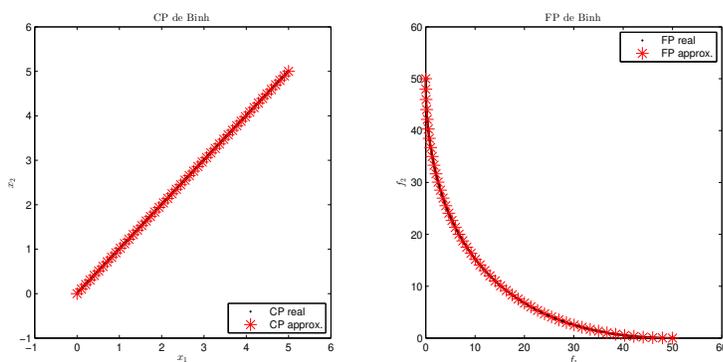
$$\sum_{i=1}^m w_i f_i(x) < \sum_{i=1}^m w_i f_i(x^*),$$

lo cual es una contradicción al supuesto que  $x^*$  es solución del problema (2.1). Por lo tanto  $x^*$  es un óptimo de Pareto.  $\square$

**Ejemplo 2.1.** Función de Binh.

$$\text{Minimizar } F = \begin{bmatrix} x_1^2 + x_2^2 \\ (x_1 - 5)^2 + (x_2 - 5)^2 \end{bmatrix}$$

con  $-5 \leq x_1, x_2 \leq 10$ . Esta función tiene un FP convexo y un CP conectado. Para elaborar este ejemplo se generaron 50 vectores de pesos  $[w_1, w_2] := [\lambda, 1 - \lambda]$  con valores de  $\lambda$  uniformemente distribuidos en el intervalo  $[0 : 1]$ . Se estableció como punto inicial el punto  $x^0 := [0, 0]$ . De la Figura 2.1 se observa que se obtienen buenos resultados ya que visualmente coinciden el FP y CP reales



**Figura 2.1:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo (se muestra el espacio de las variables de decisión del problema). En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo (espacio objetivo). Ambos resultados son de la función Binh.

marcados con  $\cdot$  color negro, y el FP y el CP obtenidos al aplicar el método, marcados con  $*$  color rojo.

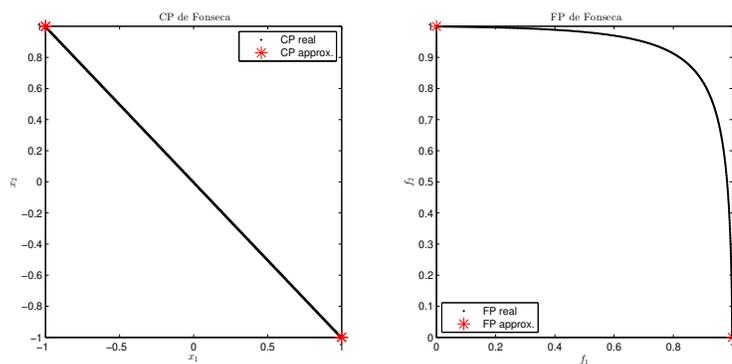
Los teoremas 2.1 y 2.2 garantizan que al resolver el problema replanteado como suma de pesos se obtienen puntos que pertenecen al conjunto solución del POM original. Sin embargo no se puede garantizar que para cada punto en el CP del POM corresponda un cierto problema de suma de pesos. En el ejemplo 2.1, el método genera todos los puntos óptimos del problema derivado de que el FP, de este caso particular, es la frontera de un conjunto convexo (a lo que comúnmente llamamos un *FP convexo*). Además de que se obtiene una buena distribución de los puntos influenciada por la distribución uniforme de los pesos. El buen funcionamiento de este método radica en la convexidad del problema multiobjetivo, puesto que el método de suma de pesos es incapaz de generar las regiones no convexas del FP; esto se puede ver en el ejemplo 2.2, y a continuación se explica el porqué de esta situación.

### Ejemplo 2.2. Función de Fonseca

$$\text{Minimizar } F(x) = \begin{bmatrix} 1 - \exp(-(x_1 - 1)^2 - (x_2 + 1)^2) \\ 1 - \exp(-(x_1 + 1)^2 - (x_2 - 1)^2) \end{bmatrix}.$$

Esta función tiene un FP no convexo y un CP conectado. Para desarrollar este ejemplo se generaron 50 vectores de pesos  $[w_1, w_2] := [\lambda, 1 - \lambda]$  con valores de  $\lambda$  uniformemente distribuidos en el intervalo  $[0 : 1]$ . Se estableció como punto inicial el punto  $x^0 := [0, 0]$ . En la Figura 2.2 pueden observar el FP y CP reales con  $\cdot$  color negro y el FP y el CP obtenidos al aplicar el método con  $*$  color rojo. En este ejemplo el método de suma de pesos solo devuelve los puntos extremos

del FP debido a la no convexidad que éste posee. Se ve claramente que el método no es útil en este caso.



**Figura 2.2:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función de Fonseca.

### 2.1.2 Discusión

Una de las ventajas más importantes de este método es que el POM se simplifica a un POE Min.  $\mathbb{R}^n \rightarrow \mathbb{R}$ . Otra ventaja es que mediante teoremas podemos garantizar que la solución obtenida con el método es un óptimo de Pareto. La principal desventaja de este método es que los puntos del FP que se localicen en una región no convexa no podrán generarse al minimizar una combinación convexa de las funciones objetivo.

#### Generación del conjunto incompleto:

Para explicar la falla en el funcionamiento del método, consideramos el caso bi-objetivo solamente. Sean  $\alpha$  y  $1-\alpha$  los pesos de los dos objetivos, con  $\alpha \in [0, 1]$ . Definimos el problema de suma de pesos como:

$$\text{Min. } (1 - \alpha)f_1 + \alpha f_2 \quad (2.2)$$

Equivalentemente,

$$\text{Min. } \frac{\cos \theta}{\cos \theta + \sin \theta} f_1 + \frac{\sin \theta}{\cos \theta + \sin \theta} f_2 \quad (2.3)$$

donde  $\theta \in [0, \frac{\pi}{2}]$ . Entonces para cierto  $\theta$ , el problema (2.2) con  $\alpha = \frac{\sin \theta}{\cos \theta + \sin \theta}$  tiene la misma solución que el problema (2.3). Luego un punto de Pareto es

solución del problema (2.2) para algún  $\alpha \in [0, 1]$  si y solo si a su vez, es solución del problema (2.3) para cierto  $\theta \in [0, \frac{\pi}{2}]$ .

Si un punto de Pareto no es solución del problema (2.3)  $\forall \theta \in [0, \frac{\pi}{2}]$  entonces no se puede obtener como solución de una combinación convexa de las dos funciones objetivo.

Considere un rotación de los ejes coordenados un ángulo  $\theta \in [0, \frac{\pi}{2}]$ , entonces aplicando la matriz de rotación a los ejes coordenados  $f_1(x)$  y  $f_2(x)$ , tenemos:

$$\begin{bmatrix} f_1(\bar{x}) \\ f_2(\bar{x}) \end{bmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix},$$

Luego, para  $f_1(\bar{x})$  tenemos

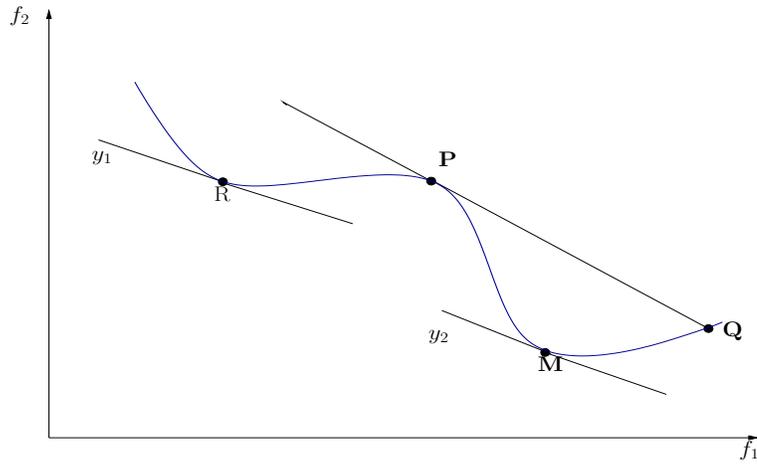
$$\frac{f_1(\bar{x})}{\cos \theta + \sin \theta} = \frac{\cos \theta f_1(x)}{\cos \theta + \sin \theta} + \frac{\sin \theta f_2(x)}{\cos \theta + \sin \theta},$$

que es el problema (2.3).

Al minimizar  $\bar{f}_1$  obtenemos un punto de Pareto. Este proceso de minimización se puede ver como trasladar el eje  $\bar{f}_2$  paralelamente a sí mismo hasta que intersecte la frontera, este punto de intersección corresponde a un punto de Pareto del problema 2.2.

Resolver el problema (2.3) para todo  $\theta \in [0, \frac{\pi}{2}]$  es equivalente a repetir el proceso de rotación para todos los ejes en  $[0, \frac{\pi}{2}]$ . Esto puede verse, como variar la pendiente de la tangente a la frontera de 0 a  $-\infty$  mientras se mantiene contacto con la frontera y se eligen los puntos de contacto. Un punto de Pareto es solución del problema (2.3) si y solo si la tangente al FP en ese punto no intersecta a la frontera en ningún otro punto donde no sea tangente a la curva. [Das and Dennis, 1997].

En la figura 2.3 se puede apreciar que es posible encontrar un punto en la región convexa del FP para el cual las rectas  $y_1$  y  $y_2$  son tangentes a él. Para la región no convexa esto no es posible, ya que la tangente al punto  $P$  intersecta al FP a su vez en el punto  $Q$ . Note que la tangente en el punto  $Q$  no coincide con la tangente del punto  $P$ . Entonces la tangente del punto  $P$  puede deslizarse aún más hacia abajo, encontrando así el punto  $Q$  que posee un valor de la función objetivo menor. Es por esto que el método de suma de pesos es incapaz de encontrar una solución en esta región.



**Figura 2.3:** Idea Geométrica del SP con un FP no convexo.

### Acerca de la distribución de las soluciones

Otra desventaja del método de SP es que una distribución uniforme de pesos no necesariamente genera una distribución uniforme de puntos sobre el FP. Lo idóneo, en la práctica, es probar con una cantidad suficiente de combinaciones de pesos. Cabe mencionar que no hay correspondencia directa entre el vector de pesos y la posición del vector solución en el espacio de los objetivos, por lo que la elección de pesos no necesariamente corresponde a que tan importante sean las funciones del vector objetivo. De esta manera el usuario no tiene control en este respecto. Esto se puede ver en los siguientes experimentos:

#### Ejemplo 2.3. Función de Poloni

$$\text{Maximizar } F(x) = \begin{bmatrix} -[1 + (A_1 - B_1)^2 + (A_2 - B_2)^2] \\ -[(x_1 + 3)^2 + (x_2 + 1)^2] \end{bmatrix}$$

con

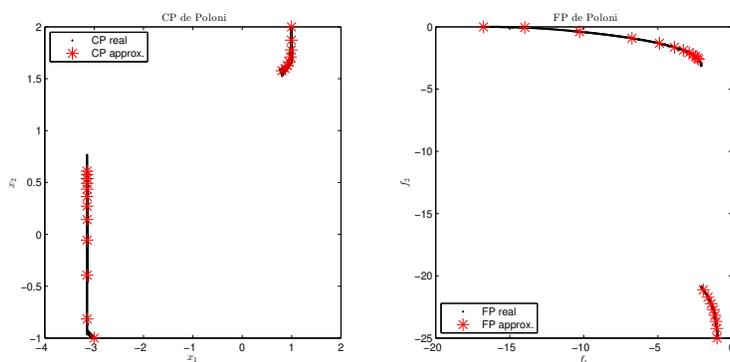
$$A_1 = 0.5 \sin 1 - 2 \cos 1 + \sin 2 - 1.5 \cos 2,$$

$$A_2 = 1.5 \sin 1 - \cos 1 + 2 \sin 2 - 0.5 \cos 2,$$

$$B_1 = 0.5 \sin x - 2 \cos x + \sin y - 1.5 \cos y,$$

$$B_2 = 1.5 \sin x - \cos x + 2 \sin y - 0.5 \cos y.$$

y  $-\pi \leq x_1, x_2 \leq \pi$ . Cabe notar que en este caso se trata de un problema de maximización, sin embargo esto no afecta al método ya que maximizar  $F$  equivale a minimizar  $-F$ ; por lo que en la práctica basta con replantearlo. El CP de este problema es no conectado al igual que el FP y ninguno parece tener una forma regular, esto debido a las restricciones que se manejan y a las combinaciones de funciones trigonométricas. Por otro lado aunque el FP no es convexo, al cambiar el problema de maximización por uno de minimización,



**Figura 2.4:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Poloni.

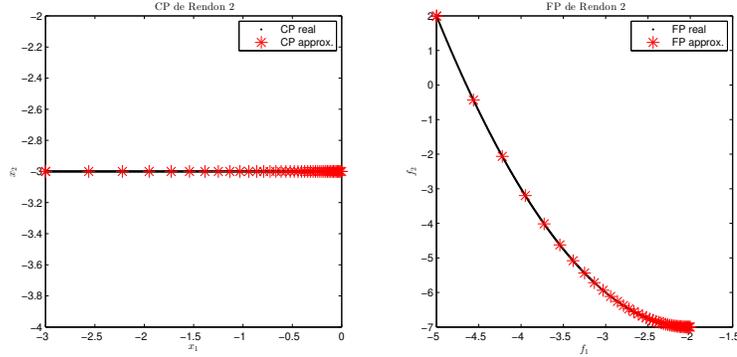
se puede pensar que si lo es. En este ejemplo se ilustra la irregularidad en la distribución de las soluciones sobre el FP, independientemente de la buena distribución de los vectores de pesos.

Para este experimento se generaron 50 pesos uniformemente distribuidos al igual que en los ejemplos anteriores. Se utilizó como punto inicial el punto  $x^0 := [0, 0]$ . En la Figura 2.4 puede observar el FP y CP reales con  $\cdot$  color negro y el FP y el CP obtenidos al aplicar el método con  $*$  color rojo.

**Ejemplo 2.4.** Función de Rendón 2:

$$\text{Min. } F(x) = \begin{bmatrix} x_1 + x_2 + 1 \\ x_1^2 + 2x_2 - 1 \end{bmatrix}$$

con  $-3 \leq x_1, x_2 \leq 3$ . Esta función tiene un FP convexo y un CP conectado. Para este experimento se generaron 50 pesos uniformemente distribuidos como en los ejemplos anteriores. Se tomó como punto inicial el punto de nuevo  $x^0 := [0, 0]$ . En la Figura 2.5 puede observar el FP y CP reales con  $\cdot$  color negro y el FP y el CP obtenidos al aplicar el método con  $*$  color rojo. Es relevante que en este ejemplo se obtiene una solución relativamente buena al problema. Se obtienen puntos en todo el FP aunque no cuenta con una buena distribución, ya que hay una acumulación de puntos en el valle del frente. Este experimento fue peculiar debido a que el CP se encuentra sobre una de las restricciones de caja, por lo que fue necesario implementar un mecanismo de manejo de restricciones para obtener los resultados mostrados.



**Figura 2.5:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Rendon 2.

### 2.1.3 Resultados numéricos

En esta sección se muestra el resumen de los resultados obtenidos en los experimentos numéricos que se desarrollaron en este trabajo para el método de SP. Las funciones de prueba utilizadas están descritas en el apéndice 6. El criterio de paro utilizado fue satisfacer la primera condición de optimalidad del POE, es decir,  $\nabla S_F(x) < 1e - 06$ . La tabla 2.1 muestra los costos en términos de evaluaciones de la función requeridos para generar todo el FP (EV.F), el número de evaluaciones de la función promedio requeridas para generar un punto (EV.F/P) y tiempo del método, para cada problema de prueba, considerando su éxito mediante la medida  $\Delta_2$ . Para cada prueba se generaron 20 puntos. Se considera que los problemas marcados en negro, en la tabla, fueron resueltos exitosamente pues los valores del indicador  $\Delta_2$  son menores a 0.01 además de la comprobación visual de los conjuntos y frentes de Pareto. En los problemas tales como *Fonseca* o *Lis* el método no tuvo resultados satisfactorios por las razones expuestas en la sección 2.1.2, en *Dent* al tener un FP con partes cóncavas y partes convexas el método tiene acumulación de soluciones en sus regiones convexas o en *ZDT3* y *Schaffer 2* se obtienen ciertos puntos en las distintas componentes del FP desconectado. Por otro lado, como era esperado las funciones como *Binh*, *Laumanns*, *Schaffer 1* y *ZDT4* obtuvieron buenos resultados al contar con frentes convexas. Una excepción interesante se da en *ZDT1*, que cuenta con un FP convexo no obtiene un buen valor de  $\Delta_2$ . Esto sucede porque las soluciones tienden hacia  $f_1(x)$ , que es la función del vector objetivo más sencilla de minimizar. Todos los ejemplos de esta tesis se ejecutaron en una PC con sistema operativo Ubuntu 14.04 LTS y procesador Intel® Core™ i3-3217U CPU 1.80GHz × 4 a 64 bits. En estos y los subsecuentes experimentos, el POE se resolvió con métodos tradicionales de optimización, tales como: Newton, Gradientes Conjugados, entre otros disponibles en la paquetería utilizada. Todos los métodos y experimentos

Tabla 2.1: Resultados numéricos del Método Suma de Pesos.

SUMA DE PESOS				
Función	# EV.F	# EV.F/P	Tiempo(segs.)	$\Delta_2$
Binh	372	18.6	0.9142	0.0479
Dent	642	32.1	1.1806	0.0269
Fonseca	540	27	0.4937	0.4024
Laumanns	178	8.8	1.0762	<b>0.0049</b>
Lis	899	44.9	2.0767	0.5015
Murata	1,029	51.4	1.6052	0.3320
Poloni	886	44.3	1.5080	0.0166
Rendón 2	574	28.7	1.1498	0.0336
Schaffer 1	116	5.8	0.3410	<b>0.0070</b>
Schaffer 2	904	45.2	2.3813	0.0962
ZDT1	9,118	455.9	53.4805	0.2205
ZDT2	8,315	415.7	3.3099	0.2206
ZDT3	17,095	854.7	6.3784	0.2233
ZDT4	23,040	1152	12.9645	<b>0.0050</b>

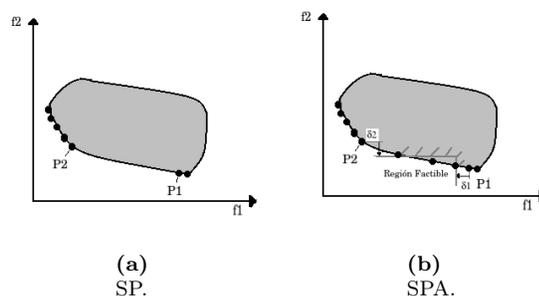
fueron programados en Matlab.

## 2.2 Método de Suma de Pesos Adaptativo

El método SPA, fue propuesto en [Kim and De Weck, 2005] y es un método alternativo basado en el método SP y que pretende subsanar las deficiencias del método original SP. La principal diferencia entre el método SP adaptativo y el original es que los vectores de pesos utilizados en el adaptativo no son predeterminados sino que evolucionan de acuerdo a la naturaleza geométrica del FP del problema.

### 2.2.1 Preliminares

**Definición 2.1.** El *punto de Nadir*  $F^{N*} \in \mathbb{R}^m$  se define como el vector que contiene los valores máximos globales individuales de cada una de las funciones objetivo pertenecientes al vector objetivo  $F^{N*} = [f_1^{N*}, \dots, f_m^{N*}]^T$ .



**Figura 2.6:** Representación del caso donde el FP es convexo con curvaturas no uniformes. La imagen (a) muestra soluciones obtenidas con SP y la imagen (b) las soluciones obtenidas a partir del SPA.

## 2.2.2 Descripción

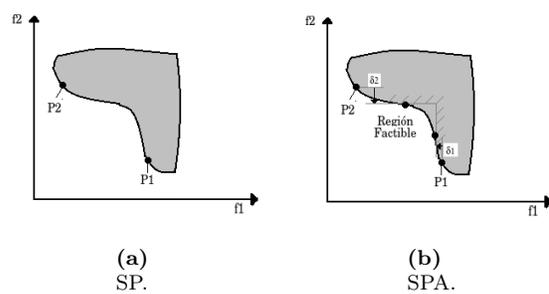
La idea central del método SPA es generar una solución preliminar del POM a partir de los resultados obtenidos mediante la aplicación del método tradicional SP, para posteriormente identificar las regiones que requieran cierto refinamiento, *i.e.*, aquellas regiones donde no se obtuvieron puntos óptimos de Pareto. Estas regiones se designan como regiones factibles para sub-optimizar. En ese momento se agregan restricciones de desigualdad en el espacio objetivo y sobre las regiones factibles y se vuelve a aplicar entonces el método SP tradicional. El proceso termina cuando todas las regiones del FP alcanzan una resolución pre-específica. Este método trata el POMs efectivamente considerando de manera especial los siguientes tres casos:

*a) Regiones convexas con curvaturas no uniformes:* en este caso el método SP obtiene la mayoría de las soluciones en las regiones con mayor curvatura del FP, se encuentran muy pocas soluciones en las regiones planas (ver la figura 2.6). Luego, el método SPA determina una región factible para refinar y así obtener más soluciones.

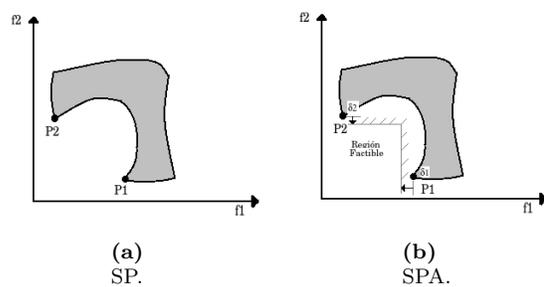
*b) Regiones no convexas con soluciones no dominadas:* el método SPA obtiene soluciones en la región no convexa al incluir restricciones de desigualdad derivadas de  $P_1$  y  $P_2$  (ver la figura 2.7) mediante las distancias  $\delta_1$  y  $\delta_2$  con dirección a los ejes coordenados respectivamente.

*c) Regiones no convexas de soluciones dominadas:* el método adaptativo evita generar puntos sobre estas regiones, a diferencia de otros métodos como el NBI (ver la sección 3.1.3) el cual genera algunas veces soluciones dominadas en estas regiones. Ver la figura 2.8.

A continuación, se describen los pasos que componen el método SPA:



**Figura 2.7:** Representación del caso donde la frontera es no convexa con soluciones no dominadas. La imagen (a) muestra soluciones obtenidas con SP y la imagen (b) las soluciones obtenidas a partir del SPA.



**Figura 2.8:** Representación del caso donde el FP es no convexo de soluciones dominadas. La imagen (a) muestra soluciones obtenidas con SP y la imagen (b) las soluciones obtenidas a partir del SPA.

1. Normalizar las funciones objetivo. Sea  $x_i^*$  el mínimo global de  $f_i(x)$ .

$$f_i = \frac{f_i - F_i^*}{F_i^{N^*} - F_i^*},$$

donde  $F^*$  es el punto de utopía y  $F^{N^*}$  es punto de Nadir.

2. Aplicar el método SP con un número pequeño de divisiones  $n_{inicial}$ . Si se utiliza un número grande de divisiones se obtienen pocas soluciones. El número de divisiones define el número de puntos deseados que se obtendrán a partir del SP.
3. Calcular el tamaño de los segmentos entre las soluciones vecinas. Eliminar las soluciones que se encuentren muy cercanas, es decir, que la distancia euclidiana sea casi cero.
4. Determinar el número de refinamientos que se necesitan en cada región. Entre más grande sea este segmento más refinamiento requiere. Este número de refinamientos se determina basado en la longitud relativa de cada segmento. Calcular entonces

$$n_i = ROUND \left( C \frac{l_i}{l_{prom}} \right),$$

donde  $n_i$  es el número de refinamientos del  $i$ -ésimo segmento,  $l_i$  es la longitud del  $i$ -ésimo segmento,  $l_{prom}$  es la longitud promedio de todos los segmentos y  $C$  es una cierta constante (parámetro del algoritmo).

5. Si  $n_i$  es menor o igual a uno, no se requiere más refinamiento. Si  $n_i$  es mayor a uno, pasar al paso 6.
6. Determinar las distancias de desplazamiento para los puntos extremos de cada segmento. Trazar la línea secante que une estos puntos extremos ( $P_1$  y  $P_2$ ). Determinar la distancia  $\delta_J$  de desplazamiento sobre el FP. Esta distancia determina la densidad final de las soluciones sobre el FP. Encontrar las distancias paralelas a los ejes objetivos, calcular el ángulo  $\theta$  como

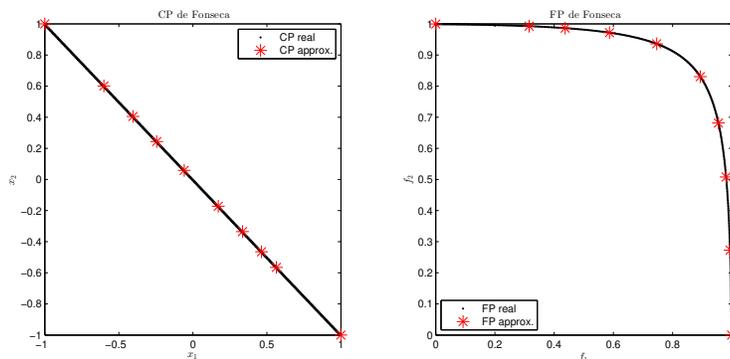
$$\theta = \arctan \left( \frac{-P_1^y - P_2^y}{P_1^x - P_2^x} \right)$$

donde  $P_i^x$  y  $P_i^y$  son las coordenadas  $(x, y)$  de los puntos  $P_1$  y  $P_2$  respectivamente. Luego calcular  $\delta_1$  y  $\delta_2$  se determinan como

$$\delta_1 = \delta_J \cos \theta \quad \text{y} \quad \delta_2 = \delta_J \sin \theta.$$

7. Agregar restricciones de desigualdad adicionales y aplicar el MSP en cada una de las regiones factibles. Los subproblemas a optimizar quedan definidos por:

$$\text{Min.} \quad \sum_{i=1}^2 w_i f_i$$



**Figura 2.9:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son para la función de Fonseca.

$$\begin{aligned} \text{s.a } f_1(x) &\leq P_1^x - \delta_1 \\ f_2(x) &\leq P_2^y - \delta_2 \end{aligned}$$

El número de divisiones del MSP lo determina  $n_i$ .

### 2.2.3 Discusión

La principal ventaja del método SPA es que logra generar soluciones en las regiones no convexas del FP. Los siguientes ejemplos ilustran este hecho.

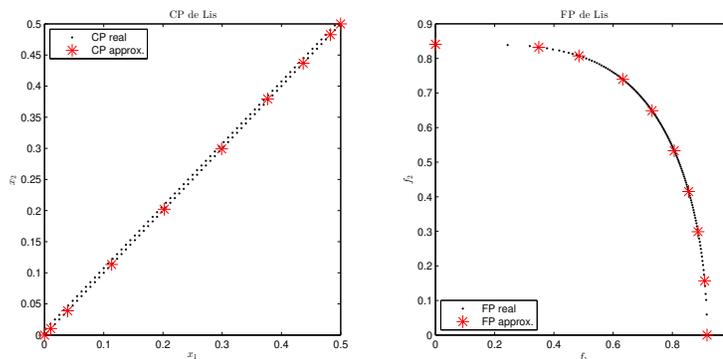
**Ejemplo 2.5.** (Función de Fonseca) Esta función se describió previamente en el ejemplo 2.2. Tiene un FP no convexo y un CP conectado. En la Figura 2.9 puede observar el FP y CP reales con . color negro y el FP y el CP obtenidos al aplicar el método con \* color rojo. Para una buena visualización se generaron únicamente 10 puntos. En este ejemplo podemos observar que, a diferencia del método SP tradicional, se obtienen puntos a lo largo de todo el FP. Cabe mencionar que dichos puntos tienen una distribución uniforme.

**Ejemplo 2.6.** (Función de Lis) Es un problema de optimización multiobjetivo, se define de la siguiente manera:

$$\text{Min. } F(x) = \left[ \sqrt[8]{x_1^2 + x_2^2}, \sqrt[4]{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} \right]^T$$

con  $-5 \leq x_1, x_2 \leq 10$ . Esta función tiene un FP no convexo y no conectado. Su CP es no conectado.

En la Figura 2.10 puede observar el FP y CP reales con . color negro y el FP y el CP obtenidos al aplicar el método con \* color rojo. Para este ejemplo



**Figura 2.10:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son para la función Lis.

se generaron 10 puntos en total. Podemos notar que el método logra generar soluciones del FP aunque este es no convexo y no conectado. Se puede apreciar una buena distribución a lo largo del FP.

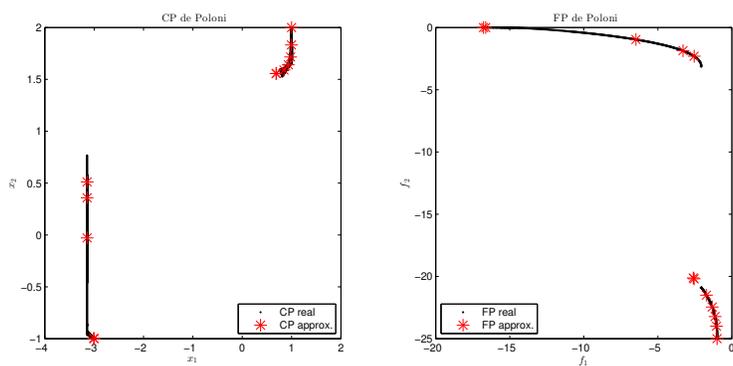
Otra ventaja del método es que gracias a que se sabe que regiones necesitan refinamiento, en teoría el método devuelve soluciones bien distribuidas. Sin embargo esto es discutible, pues en la práctica se requiere de otras consideraciones, como veremos en los siguientes ejemplos.

**Ejemplo 2.7.** (Función de Poloni) Este ejemplo fue descrito en la sección anterior, recordemos que originalmente se trata de un problema de maximización. En la figura 2.11 pueden observar el FP y CP reales con . color negro y el FP y el CP obtenidos al aplicar el método con \* color rojo. Al igual que en el ejemplo anterior se generaron 10 puntos. Observe la ventaja que tiene el SPA sobre el método NBI, del cual se hablará en la siguiente sección. Se generan soluciones sobre el FP, pero ninguna solución que sea dominada. Así mismo se puede notar la buena distribución de las soluciones obtenidas.

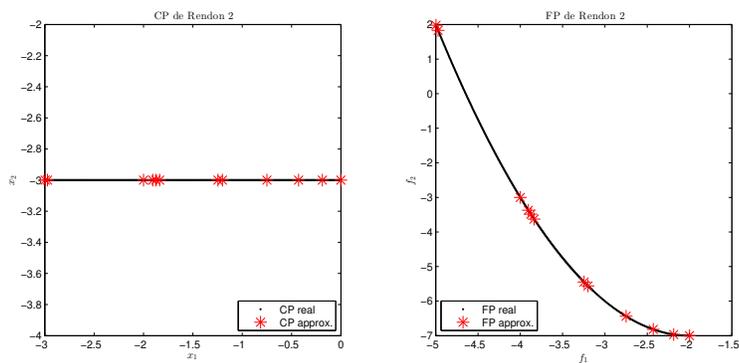
**Ejemplo 2.8.** (Función de Rendon 2) Esta función tiene un FP convexo y un CP conectado. En la Figura 2.12 puede observar el FP y CP reales con . color negro y el FP y el CP obtenidos al aplicar el método con \* color rojo. Al igual que en el ejemplo anterior se generaron 10 puntos. En este ejemplo podemos apreciar que el método genera puntos sobre el FP, aunque se recomienda pedir una mayor cantidad de puntos al realizar la ejecución, esto debido a que de vista se pueden apreciar regiones que requieren más refinamiento.

**Ejemplo 2.9.** (Función de Schaffer 2) Este es un problema que se define de la siguiente manera:

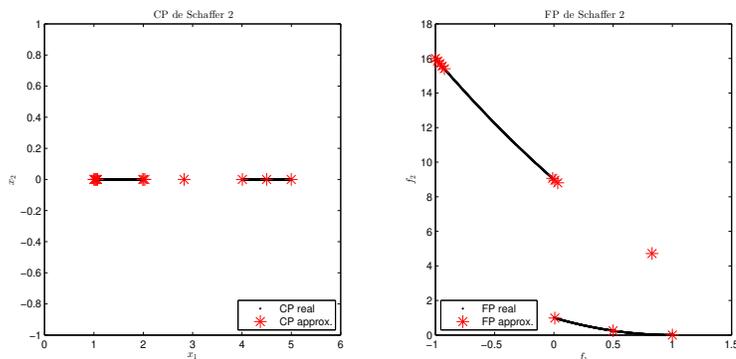
$$\text{Min. } F(x) = [ f_1(x) , f_2(x) ],$$



**Figura 2.11:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Poloni.



**Figura 2.12:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Rendon 2.



**Figura 2.13:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Schaffer 2.

donde:

$$f_1(x) = \begin{cases} -x & \text{si } x \leq 0 \\ -2 + x & \text{si } 1 < x \leq 3 \\ 4 - x & \text{si } 3 < x \leq 4 \\ -4 + x & \text{si } x > 4 \end{cases},$$

y:

$$f_2(x) = (x - 5)^2,$$

con  $-5 \leq x \leq 10$ . Esta función tiene un FP y un CP desconectado. En la Figura 2.13 puede observar el FP y CP reales con . color negro y el FP y el CP obtenidos al aplicar el método con \* color rojo. Al igual que en el ejemplo anterior se generaron 10 puntos. Podemos observar que en este ejemplo el método no tuvo buenos resultados. No logra obtener soluciones con buena distribución en la parte lineal del FP. Por otro lado al contar con un FP no conectado obtuvo un punto dominado.

Al analizar los ejemplos numéricos podemos observar que aunque se menciona un control de la distribución de soluciones, esto no es completamente cierto. El control de distribución está ligado al número de refinamientos que se solicitan pero en realidad cuando se aplica el método en dichas regiones a refinar no hay control de las distancias entre los puntos que se generan, esto sería una herencia del SP original.

Otra desventaja del método SPA es que es costoso computacionalmente, esto

sucede por la cantidad de veces que se aplica el método de SP; éste se aplica en cada región de refinamiento que requiere optimizarse. Por otro lado en problemas en regiones no convexas de soluciones dominadas, el método no genera soluciones a diferencia del método NBI que, como veremos posteriormente, podría devolver soluciones dominadas.

## 2.2.4 Experimentos numéricos

En esta sección se muestran los resultados obtenidos al aplicar el SPA a diversas funciones de prueba elegidas, algunas ya fueron descritas en las secciones anteriores, las restantes se encuentran descritas en el apéndice 6. El método se detiene hasta obtener el número deseado de puntos además de detenerse cuando se alcanza la cota máxima de evaluaciones de la función, esta se fijó en 20,000. La tabla 2.2 muestra los costos en términos de evaluaciones de la función requeridos para generar todo el FP (EV.F), el número de evaluaciones de la función promedio requeridas para generar un punto (EV.F/P) y tiempo del método, para cada problema de prueba, considerando su éxito mediante la medida  $\Delta_2$ . Para cada prueba se generaron 20 puntos. Se considera que los problemas marcados en negro, en la tabla, fueron resueltos exitosamente pues los valores del indicador  $\Delta_2$  son menores a 0.01 además de la comprobación visual de los conjuntos y frentes de Pareto. En los problemas tales como *Fonseca*, *Lis*, *Murata* se puede apreciar la ventaja principal del SPA contra el SP al lograr generar las regiones no convexas de los frentes. En los problemas como *Dent*, *Rendon 2*, *Poloni*, *Schaffer 2* vemos que las soluciones obtenidas no cuentan con una buena distribución, esto por que solo se controla el número deseado de puntos por región a sub-optimizar más no la distancia que habrá entre dichas soluciones. Por último, en las funciones con frentes convexos el resultado es exitoso al igual que el SP original.

**Tabla 2.2:** Resultados numéricos del Método Suma de Pesos Adaptativo.

SUMA DE PESOS DE PESOS ADAPTATIVO				
Función	# EV.F	# EV.F/P	Tiempo(Segs.)	$\Delta_2$
Binh	752	37.6	1.9764	0.0455
Dent	1,310	65.5	5.3817	0.0217
Fonseca	20,000	1000	38.4436	0.0179
Laumanns	688	34.4	1.5968	<b>0.0091</b>
Lis	20,000	1000	19.1231	0.0228
Murata	1,894	94.7	4.4229	0.0798
Poloni	3,102	155.1	3.6954	0.1783
Rendón 2	895	44.7	1.8174	0.0195
Schaffer 1	455	22.7	3.0412	<b>0.0051</b>
Schaffer 2	4,855	242.7	8.0983	0.3191
ZDT1	20,000	1000	21.4833	0.4900
ZDT2	20,000	1000	32.4464	1.1028
ZDT3	20,000	1000	18.2943	0.4413
ZDT4	20,000	1000	19.2491	<b>0.0031</b>

## Capítulo 3

# Métodos de punto de referencia y continuación

### 3.1 Método NBI

El método de *Intersección de la Dirección Normal con la Frontera*, o NBI fue propuesto en 1998 por J.Dass y J.E Dennis [Das and Dennis, 1998]. Este método aprovecha el hecho que el FP de cualquier POM se encuentra en el conjunto frontera de la imagen bajo las funciones objetivo a optimizar. Este método encuentra puntos candidatos a óptimos de Pareto mediante la solución de una serie de POEs asociados. El método NBI pertenece a la clase de métodos conocidos como de *punto de referencia* pues utiliza un cierto punto (inalcanzable) para guiar la solución de cada uno de los POEs asociados. Otros métodos de punto de referencia pueden consultarse en [Ehrgott, 2005, Figueira et al., 2005].

#### 3.1.1 Preliminares

Antes de describir el método NBI, se darán unas definiciones necesarias para la construcción del punto de referencia a utilizar. Recordemos que queremos minimizar  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , y supongamos entonces que para todo  $i \in \{1, \dots, m\}$  es posible encontrar un vector

$$x_i^* \in \mathbb{R}^n \tag{3.1}$$

tal que  $x_i^*$  es un mínimo global de  $f_i$ , con  $f_i^* := f_i(x_i^*)$  el valor óptimo de la  $i$ -ésima función objetivo.

**Definición 3.1.** *El punto ideal, de utopía, o vector utópico  $F^* \in \mathbb{R}^m$  se define como el vector formado por los valores mínimos globales, individuales, de cada*

una de las funciones objetivo. Es decir

$$F^* = [f_1^*, \dots, f_m^*]^T. \quad (3.2)$$

**Definición 3.2.** Sea  $F_i^* = F(x_i^*)$  con  $x_i^*$  definido en 3.1 para  $i \in \{1, \dots, m\}$ . Sea  $\Phi$  la matriz de tamaño  $m \times m$ , cuya  $i$ -ésima columna es  $F_i^* - F^*$ . Dicha matriz es conocida como la matriz de pago del POM. Definiremos como la **CHIM**, por sus siglas en inglés *Convex Hull of Individual Minima*, al conjunto de puntos en  $\mathbb{R}^m$  que son combinación convexa de las columnas de  $\Phi$ , es decir,

$$CHIM = \{\Phi\beta : \beta \in \mathbb{R}^m, \sum_i \beta_i = 1, \beta_i \geq 0\}$$

### 3.1.2 Descripción

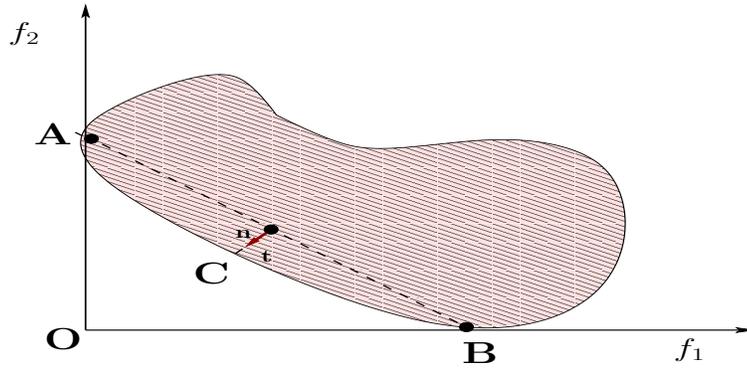
Como se mencionó anteriormente, el método NBI se centra en el espacio objetivo. Este método utiliza el punto ideal como punto de referencia. Cabe mencionar que no es necesario tener el valor exacto del punto ideal; con una buena aproximación de éste es suficiente para obtener buenos resultados. La CHIM es una forma de caracterizar el conjunto convexo más pequeño que contiene los valores mínimos de las funciones objetivo por separado.

Para iniciar el método es necesario realizar una traslación de la forma  $F(x) \leftarrow F(x) - F^*$ . Al hacer esta traslación<sup>1</sup> se puede trabajar en un espacio donde la frontera buscada se encuentra inmersa en un cuadrante con coordenadas no negativas, donde el punto ideal es el origen y las coordenadas de los vectores columna de la matriz de pago están sobre los planos que forman los ejes del nuevo sistema de coordenadas. Posteriormente se toma un cierto punto sobre la CHIM, digamos  $\Phi\beta$ , y se calcula el vector  $\hat{n}$  unitario normal a la CHIM, que pasa por  $\Phi\beta$  con dirección hacia el origen. Entonces,  $\Phi\beta + t\hat{n}$ , con  $t \in \mathbb{R}$  describe el conjunto de puntos pertenecientes a la recta normal construida. El punto de intersección de dicha recta normal con la frontera del POM se encuentra al resolver el siguiente POE con restricciones:

$$\begin{aligned} & \text{Maximizar}_{x,t} && t \\ & \text{sujeto a} && F(x) \leq \Phi\beta + t\hat{n}. \end{aligned} \quad (3.3)$$

Cada vector  $\beta$  puede verse como un parámetro que define un cierto problema (3.3) particular, que se denota como el subproblema  $NBI_\beta$ . Lo que se obtiene al aplicar el método NBI es la solución de los diversos subproblemas  $NBI_\beta$  para distintos valores de  $\beta$ . Dichos valores se toman idealmente bien

<sup>1</sup>Dependiendo de los valores de la función, el vector ideal podría sumarse en vez de sustraerse.



**Figura 3.1:** Esquema del funcionamiento del método NBI. Donde  $O$  es el punto ideal,  $A$  corresponde a  $F_1^*$ ,  $B$  corresponde a  $F_2^*$ ,  $AB$  define la CHIM, el arco  $ACB$  es el FP del problema,  $n$  es la dirección normal y  $t$  es el valor óptimo de (3.3) (la distancia máxima).

distribuidos sobre la CHIM, buscando una buena distribución de las soluciones sobre el FP. La restricción de desigualdad de (3.3) asegura que el punto solución  $x$  es asignado bajo  $F$  a un punto en la recta normal. En [Motta et al., 2012] se hace notar que la condición de desigualdad de los  $NBI_\beta$  logra una mayor libertad del algoritmo para encontrar óptimos globales y evita malas formulaciones de dichos subproblemas.

La Figura 3.1 muestra la idea geométrica detrás del método. Al realizar la traslación, el origen del plano coordenado corresponde al punto ideal. El punto  $A$  denota a  $F_1^*$ , el punto  $B$  denota a  $F_2^*$ ,  $AB$  conforma en este caso la CHIM, el arco  $ACB$  es el FP para este ejemplo,  $n$  es la dirección normal en el punto  $(\beta, \Phi\beta)$  y la distancia  $t$  es el parámetro a maximizar.

Por las características de la traslación del origen realizada para el método, el punto  $x^*$  obtenido para el máximo  $t$  resulta ser el mismo tanto en el espacio coordenado como en el problema original. Por la forma en que se definen los subproblemas  $NBI_\beta$ , es posible encontrar el conjunto de puntos cuya imagen en el espacio objetivo es la frontera de la superficie sin importar si ésta es convexa o no convexa. También, entre más subproblemas  $NBI_\beta$  se resuelvan más puntos sobre la frontera se tendrán. Aunque como se verá con los ejemplos posteriores, estos puntos no necesariamente son óptimos de Pareto.

En resumen, la idea central del método NBI es que el punto de intersección entre la frontera de la imagen y la normal que apunta hacia el origen saliendo desde cualquier punto de la CHIM es un punto perteneciente a una cierta porción de la frontera que contiene a las imágenes de los puntos óptimos de Pareto.

### Aplicaciones exitosas

Este método ha sido empleado en diversas áreas de la ingeniería. Por ejemplo, en [Motta et al., 2012] se empleó el método para resolver un problema biobjetivo de ingeniería química de un reactor isotérmico discontinuo. En este reactor se lleva a cabo cierta reacción química. Se busca maximizar el rendimiento de una cierta sustancia y reducir el costo de la separación de las sustancias empleadas en la reacción.

Otro ejemplo se muestra en [Bhaskar et al., 2000], donde se emplea el NBI para optimizar una caja de cambios. La meta era minimizar el volumen del reductor de velocidad y la presión que se ejerce en dos de los ejes de la caja.

### 3.1.3 Discusión

Existe una relación entre los subproblemas del NBI y el problema de SP. Esta relación muestra cómo ir y venir entre el parámetro  $\beta$  que define al subproblema NBI y una combinación convexa del vector de pesos  $w$  correspondiente a un punto óptimo de Pareto particular. Así mismo esta relación muestra que para cada  $w$  existe un  $\beta$  tal que un subproblema  $NBI_\beta$  tiene la misma solución que un problema  $SP_w$ , pero al revés esto no se cumple.

Dado un cierto punto óptimo de Pareto  $x^*$ , definimos el problema de Suma de Pesos como:

Sea  $w \in [0, 1]^m$  y  $\sum_{i=1}^m w_i = 1$ .

$$\text{Min}_x w^T F(x) \tag{3.4}$$

Sea  $x^*$  la solución del problema (3.4), por la primera parte de las condiciones de KKT para optimalidad tenemos que:

$$\nabla_x F(x^*)^T w = 0 \tag{3.5}$$

Sea  $\beta$  el vector de parámetros en  $NBI_\beta$ . Definimos el subproblema  $NBI_\beta$  como:

$$\begin{aligned} & \text{Min}_{x,t} -t & (3.6) \\ \text{s.a } & F(x) - \Phi\beta - t\hat{n} = 0. & (3.7) \end{aligned}$$

Sea  $(x^*, t^*, \lambda^{(1)*})$  la solución del problema (3.6). Por la primera parte de las condiciones de KKT para optimalidad tenemos que:

$$\begin{aligned}\nabla_x F(x^*)^T \lambda^{(1)*} &= 0 \\ -1 + \hat{n}^T \lambda^{(1)*} &= 0\end{aligned}\tag{3.8}$$

donde  $\lambda^{(1)*} \in \mathbb{R}^n$  representa el vector de multiplicadores correspondiente a la restricción  $\Phi\beta + t\hat{n} - F(x) = 0$ .

**Observación 3.1.** Suponga  $(x^*, t^*, \lambda^{(1)*})$  es la solución de  $NBI_\beta$  y que  $\sum_{i=1}^m \lambda_i^{(1)*} \neq 0$ . Ahora definimos las componentes del vector  $w$  como

$$w_i = \frac{\lambda_i^{(1)*}}{\sum_{i=1}^m \lambda_i^{(1)*}}.$$

Luego el problema (3.4) con el vector de pesos  $w$  anterior tiene como solución a  $x^*$ .

**Demostración 3.1.** Dividiendo ambos lados de (3.8) por el escalar  $\sum_{i=1}^m \lambda_i^{(1)*}$  tenemos

$$\frac{\nabla_x F(x^*)^T \lambda^{(1)*}}{\sum_{i=1}^m \lambda_i^{(1)*}} = 0,$$

es clara la equivalencia entre (3.5) y (3.8). Es importante notar que si para algún  $i$ , el signo de  $\lambda_i^{(1)*}$  es contrario al de  $\sum_{i=1}^m \lambda_i^{(1)*}$ , entonces el vector  $w$  tiene una componente negativa y no califica como un peso posible para el problema (3.4). En tal caso, o la optimalidad de Pareto del punto  $NBI(x^*, t^*, \lambda^{(1)*})$  es cuestionable o el punto de Pareto se encuentra en una región no convexa, por lo que no existe una combinación convexa correspondiente en el problema de suma de pesos.

**Observación 3.2.** Supongamos que  $x^*$  resuelve el problema  $SP_w$ . Sea  $(\bar{\beta}, t^*)$  la solución del  $(m+1) \times (m+1)$  sistema lineal

$$\begin{aligned}\Phi\bar{\beta} + t^*\hat{n} &= F(x^*), \\ \sum_i \beta_i &= 1.\end{aligned}$$

Luego  $x^*$  corresponde a la solución de  $NBI_\beta$  con  $\bar{\beta} = \beta$ , la solución del problema (3.6) es:

$$\left[ x^*, t^*, \lambda^{(1)*} = \frac{w}{w^T \hat{n}} \right].$$

**Demostración 3.2.** *Dividiendo ambos lados de (3.5) por  $w^T \hat{n}$ . Esto se puede realizar ya que  $w_i \geq 0$  y  $\hat{n}$  tiene componentes negativas, de modo que  $w^T \hat{n} < 0$ . Tenemos entonces que:*

$$\frac{\nabla_x F(x^*)^T w = 0}{w^T \hat{n}} = 0.$$

Y observe que

$$\hat{n}^T \lambda^{(1)*} = \hat{n}^T \left( \frac{w}{w^T \hat{n}} \right) = 1,$$

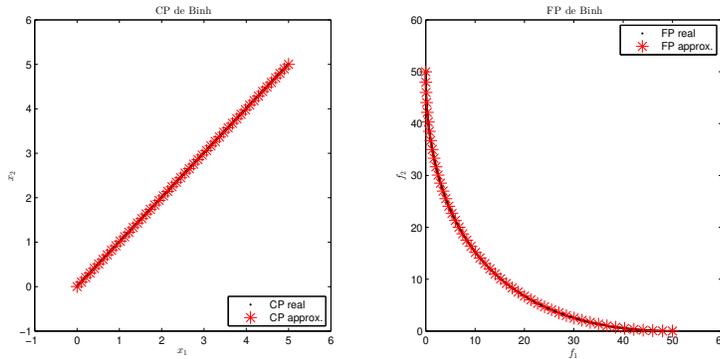
entonces la primera parte de las condiciones de KKT del  $NBI_\beta$  se conservan. Luego note que  $\Phi\beta + t\hat{n} = F(x^*)$ . Entonces es clara la equivalencia entre  $SP_w$  y  $NBI_\beta$ .

### Distribución de las soluciones en regiones convexas y no convexas:

Una de las ventajas de este método es que el POM es simplificado a un POE, el cual se resuelve con métodos tradicionales de optimización con restricciones. Así mismo, con este método se logra generar puntos tanto las regiones convexas como las regiones no convexas del FP. Una ventaja notable es que, a diferencia de los métodos agregativos, con el NBI es posible tener un cierto control sobre la distribución de los puntos generados sobre el FP. Veamos los resultados obtenidos para los siguientes experimentos:

**Ejemplo 3.1.** (Función de Binh) Este problema ya ha sido descrito en el capítulo anterior, recordemos que en este caso se tiene un FP convexo y un CP conectado. En la Figura 3.2 se marcan el FP y CP teóricos del problema con . color negro y el FP y el CP obtenidos al aplicar el método con \* color rojo. Para este experimento se generaron 50 puntos uniformemente distribuidos sobre la CHIM. En este ejemplo podemos apreciar como el método NBI devuelve resultados muy buenos para este tipo de problemas, en los que el FP es convexo. Podemos observar que los puntos se encuentran sobre el frente y además poseen una buena distribución. Se esperaban resultados buenos por la naturaleza del POM particular.

**Ejemplo 3.2.** (Función de Fonseca) Este problema ya ha sido descrito en el capítulo anterior, recordemos que para esta función se tiene un FP no convexo y un CP conectado. En la Figura 3.3 pueden observar el FP y CP teóricos del problema con . color negro y el FP y el CP obtenidos al aplicar el método con \* color rojo. Para este experimento se generaron 50 puntos uniformemente distribuidos sobre la CHIM. En este ejemplo podemos observar que el NBI devuelve de nuevo muy buenos resultados. Se logra generar puntos para describir todo el FP a pesar de ser no convexo, a diferencia de lo que sucedía en esta misma



**Figura 3.2:** En la primera imagen se puede observar el CP teórico en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP teórico en color negro y FP aproximado en color rojo. Ambos resultados son para la función de Binh.

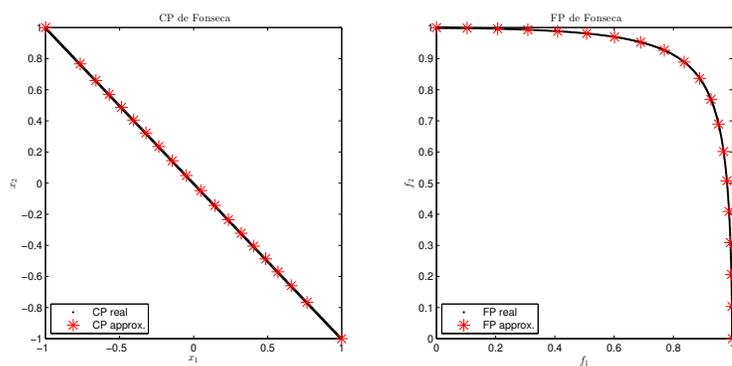
función aplicando el SP. Por otro lado, cabe destacar que los puntos presentan una muy buena distribución sobre el FP.

### Generación de puntos no óptimos:

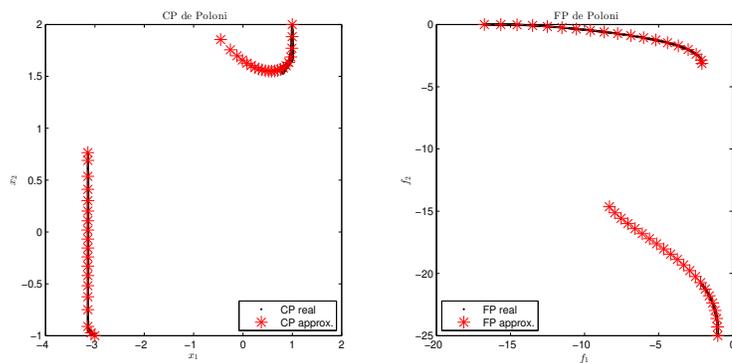
Un primer inconveniente del método NBI es que no se puede garantizar que las soluciones obtenidas de los subproblemas derivados sean óptimos de Pareto; en especial para FPs con componentes desconectadas. Es decir, el método solo garantiza que se encuentran puntos en la frontera pero estos podrían ser dominados. Los siguientes experimentos hacen notar esta deficiencia:

**Ejemplo 3.3.** (Función de Poloni) Definido previamente, este POM corresponde a un problema de maximización. En este caso se tiene un FP no conectado, al igual que el CP. En la Figura 3.4 se pueden observar el FP y CP teóricos con  $\cdot$  color negro y con  $*$  color rojo los conjuntos aproximados FP y CP obtenidos al aplicar el método. Se generaron, para este experimento, 50 puntos uniformemente distribuidos sobre la CHIM. Se puede observar una de las principales desventajas del método; vemos que se obtienen puntos dominados como solución de algunos subproblemas. Estos puntos a pesar de estar en la frontera, no necesariamente corresponden a óptimos de Pareto.

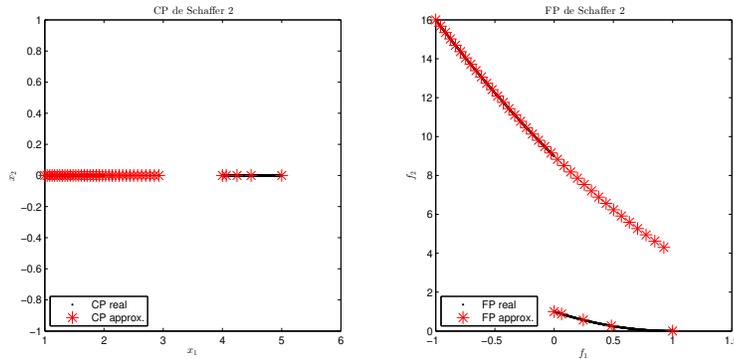
La desventaja del método NBI ilustrada en el ejemplo anterior podría superarse, si se resuelven suficientes subproblemas y se hace al final un filtrado para descartar aquellos puntos obtenidos que sean dominados. Sin embargo esto depende de que efectivamente se logren generar puntos suficientes; además de que el hecho de tener que resolver demasiados subproblemas para después descartar



**Figura 3.3:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son para la función de Fonseca.



**Figura 3.4:** En la primera imagen se puede observar el CP teórico en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función de Poloni.



**Figura 3.5:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Schaffer 2.

soluciones no es del todo eficiente. En las figuras correspondientes al ejemplo 3.3 podemos apreciar que en la parte del FP donde los puntos devueltos por el NBI sí son solución del problema multiobjetivo se cuenta con una buena distribución de éstos sobre el frente. El siguiente ejemplo muestra que no necesariamente es así en todos los casos.

#### Fallas de distribución de soluciones en componentes de con curvaturas dispares:

**Ejemplo 3.4.** (Función de Schaffer 2) Este problema presenta un FP y un CP desconectado. En la Figura 3.5 puede observar el FP y CP exactos con  $\cdot$  color negro y el FP y el CP obtenidos al aplicar el método con  $*$  color rojo. Se generaron 50 puntos uniformemente distribuidos sobre la CHIM. En este experimento vuelve a ocurrir que el método NBI devuelve puntos en la frontera que no son óptimos de Pareto. Una peculiaridad más de este ejemplo es que los puntos óptimos de Pareto no están distribuidos uniformemente sobre las componentes. Por la construcción geométrica del método es claro que la distribución se ve afectada por las disparidades de curvatura en las dos diferentes componentes del {FP}.

#### Funciones objetivo multimodales:

Otra desventaja importante de este método, y que afecta tanto en eficiencia como en distribución de las soluciones es que si al calcular el valor óptimo  $f_i^*$  de alguna de las funciones objetivo por separado no se alcanza el óptimo

global<sup>2</sup> entonces se generan diferentes conjuntos CHIM. Esto produce resultados distintos según los valores que son considerados; como se verá en los ejemplos siguientes.

**Ejemplo 3.5.** (Función ZDT1 [Zitzler et al., 2000]) El siguiente POM trabaja con 30 variables y este definido como

$$\text{Min. } F = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix}$$

donde

$$f_1(x) = x_1; \quad f_2(x) = g(x) * \left(1 - \sqrt{\frac{1}{g(x)}}\right),$$

y

$$g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^{30} x_i,$$

con  $0 \leq x_i \leq 1$  para  $i \in \{1, \dots, 30\}$ . Este POM posee un FP convexo. Lo peculiar de este problema es que el punto óptimo de  $f_1(x)$  no es único, de hecho existe una infinidad de puntos que alcanzan el mínimo global. Esto trae como consecuencia que se obtenga una CHIM distinta, dependiendo del punto que se considere para su construcción. Es sencillo<sup>3</sup> ver que los vectores de la forma

$$x^{*1} = [0, x_2, \dots, x_{30}] \quad (3.9)$$

con  $x_i \in [0, 1]$  para  $i = 2, \dots, 30$  son mínimos globales de  $f_1$ , donde  $f_1(x^{*1}) = 0$ . Por otro lado la función  $f_2$  tiene un único vector óptimo, que es  $x^{*2} = [1, 0, \dots, 0]$  con valor óptimo  $f_2(x^{*2}) = 0$ .

Así pues, para este experimento se tomaron 3 puntos distintos para  $x^{*1}$  de la forma

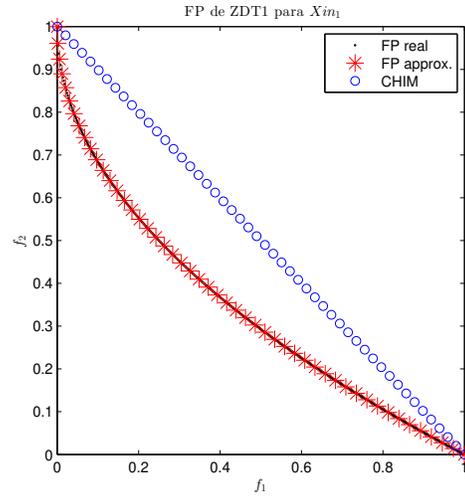
$$\begin{aligned} x_1^{*1} &= [0, x_2, \dots, x_{30}], \quad x_i = 0.0 \quad \forall i \in \{2, \dots, 30\}, \\ x_2^{*1} &= [0, x_2, \dots, x_{30}], \quad x_i = 0.25 \quad \forall i \in \{2, \dots, 30\}, \\ x_3^{*1} &= [0, x_2, \dots, x_{30}], \quad x_i = 0.5 \quad \forall i \in \{2, \dots, 30\}, \end{aligned}$$

y se generaron 50 puntos uniformemente distribuidos sobre cada una de las CHIM correspondientes. Observe que el punto ideal es  $F^* = [0, 0]^T$  y no se necesita realizar la traslación al origen.

Aplicando el NBI a este POM obtenemos distintos puntos solución, dependiendo la CHIM considerada. Esto se puede apreciar en la Figura 3.6, Figura 3.7

<sup>2</sup>La mayoría de los métodos matemáticos solo garantizan optimalidad local.

<sup>3</sup>Analizando la función  $f_1$ , vemos que esta definida únicamente por el primer componente del vector  $x$  que se evalúa. De modo que todos aquellos vectores que tengan el mismo valor en el primer componente poseen la misma imagen. Debido a la restricción de caja, el valor mínimo posible es cero. Al tratarse de un problema de minimización y considerando las restricciones de caja tenemos que todos los vectores mínimos deben ser de la forma (3.9).

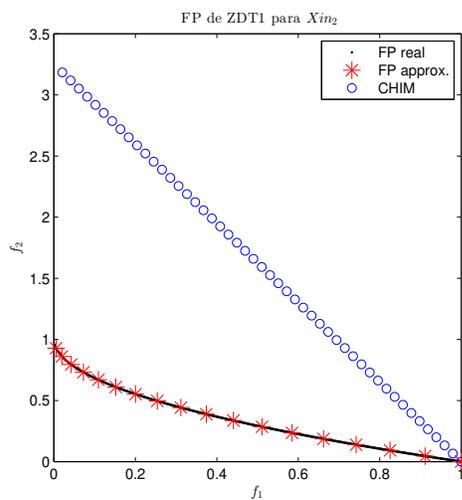


**Figura 3.6:** En la imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Resultado para la función ZDT1.

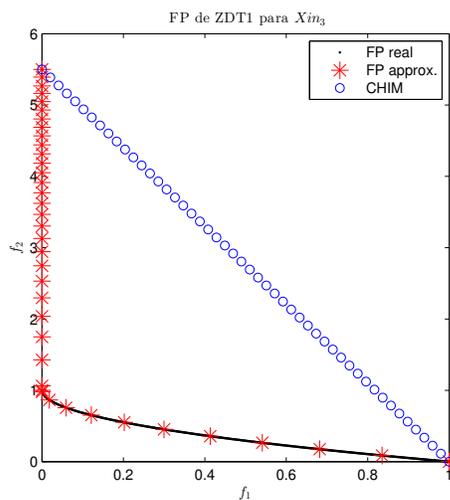
y la figura 3.8. Los FPs verdaderos están marcados con  $\cdot$  color negro y el FP obtenido al aplicar el método en cada uno de los casos está marcado con  $*$  color rojo. Finalmente, podemos observar que entre más cercano sea el punto inicial al vector cero, mejor solución se obtendrá.

### 3.1.4 Experimentos numéricos

En esta sección se muestra el resumen de los resultados obtenidos en los experimentos numéricos que se desarrollaron en este trabajo para el método de NBI. Las funciones de prueba utilizadas están descritas en el apéndice 6. El criterio de paro utilizado fue satisfacer la primera condición de optimalidad del subproblema  $NBI_{\beta}$ . La tabla 3.1 muestra los costos en términos de evaluaciones de la función requeridos para generar todo el FP (EV.F), el número de evaluaciones de la función promedio requeridas para generar un punto (EV.F/P) y tiempo del método, para cada problema de prueba, considerando su éxito mediante la medida  $\Delta_2$ . Para cada prueba se generaron 20 puntos. Se considera que los problemas marcados en negro, en la tabla, fueron resueltos exitosamente pues los valores del indicador  $\Delta_2$  son menores a 0.01 además de la comprobación visual de los conjuntos y frentes de Pareto. Observe que en 9 de las 14 funciones de prueba seleccionadas el método devuelve buenos resultados en el valor de  $\Delta_2$ , observe que en estos casos se trata de frentes convexos, no convexos y mixtos. La característica que poseen en común es que son frentes de Pareto conectados como: *Dent*, *Murata* y *Schaffer 1*. Note que en las funciones que cuentan con



**Figura 3.7:** En la imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Resultado para la función ZDT1.



**Figura 3.8:** En la imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Resultado para la función ZDT1.

Tabla 3.1: Resultados numéricos del Método NBI.

NBI				
Función	# EV.F	# EV.F/P	Tiempo(segs.)	$\Delta_2$
Binh	488	24.4	0.7407	0.0479
Dent	516	25.8	0.9181	<b>0.0053</b>
Fonseca	954	47.7	2.7747	<b>0.0004</b>
Laumanns	788	39.4	1.8826	<b>0.0049</b>
Lis	1,737	86.8	1.9402	0.0133
Murata	316	15.8	0.7634	<b>0.0017</b>
Poloni	868	43.4	1.5558	0.6885
Rendon 2	462	23.1	1.0037	<b>0.0070</b>
Schaffer 1	239	11.9	0.7662	<b>0.0071</b>
Schaffer 2	808	40.4	2.3199	0.2877
ZDT1	5821	291.1	6.5643	<b>0.0009</b>
ZDT2	4,058	202.9	3.8568	<b>0.0045</b>
ZDT3	6,743	337.1	7.7489	0.0240
ZDT4	4,910	245.5	3.1006	<b>0.0020</b>

frentes no conectados como *Poloni*, *Schaffer 2* y *ZDT3* el valor del indicador  $\Delta_2$  no es cercano a cero. Esto ocurre puesto que el indicador de desempeño penaliza al método cuando genera puntos que no son parte del FP sino que son puntos pertenecientes a la frontera.

## 3.2 Método Pareto Tracer

### 3.2.1 Preliminares

Los métodos de *Continuación Numérica* consideran curvas que están implícitamente definidas mediante sistemas de ecuaciones no lineales indeterminados. El problema está dado por

$$H(x, \lambda) = 0, \quad (3.10)$$

donde  $H : \mathbb{R}^{N+1} \rightarrow \mathbb{R}^N$  es suave,  $x \in \mathbb{R}^N$  y  $\lambda \in \mathbb{R}$ .

**Teorema 3.1. Teorema de la Función Implícita.** Sean  $G_k \in \mathbb{R}^k$ ,  $G_N \in \mathbb{R}^N$  conjuntos no vacíos y abiertos. Sea  $H : G_k \times G_N \rightarrow \mathbb{R}^N$  continuamente diferenciable. Y sea  $x_0 \in G_k$ ,  $y_0 \in G_N$  tales que

$$H(x_0, y_0) = 0,$$

$y$

$$\frac{\partial H}{\partial y}(x_0, y_0) \text{ regular.}$$

Entonces existen subconjuntos  $\tilde{G}_k \subset G_k$ ,  $\tilde{G}_N \subset G_N$  y exactamente una función continua

$$f : \tilde{G}_k \rightarrow \tilde{G}_N$$

con  $f(x_0) = y_0$  y  $H(x, f(x)) = 0 \forall x \in \tilde{G}_k$ . Por lo que  $f(x)$  es la única solución de  $H(x, y) = 0$  dentro de  $\tilde{G}_N$ .

*Demostración.* Definimos la función

$$\begin{aligned} g : G_k \times G_N &\rightarrow \mathbb{R}^{k+N} \\ (x, y) &\rightarrow g(x, y) = (x, H(x, y)). \end{aligned} \quad (3.11)$$

Evidentemente  $g$  es continuamente diferenciable puesto que  $H$  lo es. El jacobiano de  $g$  en  $(x_0, y_0) \in G_k \times G_N$  es

$$\begin{bmatrix} I_{k \times k} & 0 \\ \frac{\partial H}{\partial x}(x_0, y_0) & \frac{\partial H}{\partial y}(x_0, y_0) \end{bmatrix}, \quad (3.12)$$

es regular dado que  $\det \left( \frac{\partial H}{\partial y}(x_0, y_0) \right) \neq 0$ . Además  $g(x_0, y_0) = (x_0, H(x_0, y_0)) = (x_0, 0)$ . Aplicando el teorema de la función inversa tenemos que  $\tilde{G}_k \in G_k \times G_N$  y  $\tilde{G}_N \in \mathbb{R}^{k+N}$  tales que  $(x_0, y_0) \in \tilde{G}_k$ ,  $(x_0, 0) \in \tilde{G}_N$  y  $g : \tilde{G}_k \rightarrow \tilde{G}_N$  es una función biyectiva con inversa continuamente diferenciable.

Definimos  $W := \{z \in \mathbb{R}^k \mid (z, 0) \in \tilde{G}_N\}$ , entonces  $x_0 \in W$ . Luego para cada  $z \in W$

$$g(x, y) = (z, 0) \quad (3.13)$$

tiene un único par  $(x_z, y_z)$  como solución, *i.e.*,

$$(x_z, H(x_z, y_z)) = (z, 0) \Rightarrow x_z = z. \quad (3.14)$$

Por lo tanto  $H(z, y_z)$ . Note que la pre-imagen  $g^{-1}(W)$  es cerrada en  $G_k \times G_N$ . De modo que el interior de la pre-imagen  $g^{-1}(W)^c$  es abierto en  $\tilde{G}_k \times \tilde{G}_N$  donde  $\tilde{G}_k \subset G_k$  y  $\tilde{G}_N \subset G_N$  son abiertos. Además  $H(x, y) = 0$  con  $(x, y) \in \tilde{G}_k \times \tilde{G}_N$ .

Ahora como  $y_z$  es único dado  $z$ , definimos una función  $f(z) = y_z$  donde  $z \in \tilde{G}_k$  y  $y_z \in \tilde{G}_N$ . Luego  $f(x_0) = y_0$  y

$$(z, y_z) = g^{-1}(z, 0) = (z, f(z)). \quad (3.15)$$

Por lo que  $H(z, f(z)) = 0$ . De esta forma  $f$  es la función que estamos buscando.  $\square$

Por el teorema 3.1 se sigue que el conjunto de solución de (3.10), *i.e.*,

$$H^{-1}(0) = \{(x, \lambda) \in \mathbb{R}^{N+1} : H(x, \lambda) = 0\} \quad (3.16)$$

es, al menos localmente y bajo cierto supuestos, una curva. Dentro de los métodos de continuación numérica encontramos los métodos del tipo *Predictor-Corrector* (PC). En lo posterior explicaremos los pasos principales de dichos métodos para trazar una curva uni-dimensional. Consideramos (3.10), y que se tiene un punto  $x_0 \in \mathbb{R}^{N+1}$  con  $H(x_0) = 0$  y el *rango*( $H(x_0)$ ) =  $N$ . Por el teorema 3.1 existe una curva  $c : (0 - \varepsilon, 0 + \varepsilon) \rightarrow \mathbb{R}^{N+1}$  con  $c(0) = x_0$  y

$$H(c(s)) = 0 \quad \forall s \in (0 - \varepsilon, 0 + \varepsilon). \quad (3.17)$$

Diferenciando (3.17) obtenemos

$$H'(c(s)) * c'(s) = 0. \quad (3.18)$$

Por lo tanto, los vectores tangentes  $c'(s)$  (y en consecuencia, la linearización de la curva en  $x = c(s)$ ) pueden encontrarse calculando el núcleo de  $H'(x)$ . Esto se realiza mediante una descomposición QR de  $H'(x)^T$ : si

$$H'(x)^T = QR \quad (3.19)$$

para un matriz ortogonal  $Q \in \mathbb{R}^{(N+1) \times (N+1)}$  y una matriz triangular superior  $R \in \mathbb{R}^{(N+1) \times N}$ , entonces la última columna del vector  $q_{N+1}$  de  $Q$  es el núcleo deseado. La orientación de la curva puede calcularse al monitorear el signo de

$$\det \begin{pmatrix} H'(x) \\ q_{N+1}^T \end{pmatrix}. \quad (3.20)$$

Una vez obtenido el vector que apunta a lo largo de la curva solución linearizada, se puede realizar un movimiento en dicha dirección lo que devuelve una solución del predictor  $p$ . Para la solución del corrector, uno puede regresar a  $c$  al aplicar el método de Gauss-Newton a (3.10) tomando como punto inicial a  $p$ . De este modo, es posible calcular un secuencia de soluciones que estarán

alineadas a lo largo de la curva solución  $H^{-1}(0)$ . Este caso se puede extender al caso multiobjetivo al considerar  $\tilde{F} : \mathbb{R}^{n+k} \rightarrow \mathbb{R}^{n+1}$

$$\tilde{F}(x, \alpha) = \begin{pmatrix} \sum_{i=1}^k \alpha_i \nabla f_i(x) \\ \sum_{i=1}^k \alpha_i - 1 \end{pmatrix} = 0. \quad (3.21)$$

El conjunto de puntos de KKT está contenido en el conjunto cero de  $\tilde{F}$ , lo que motiva la continuación de  $\tilde{F}^{-1}(0)$ .

### 3.2.2 Descripción

A continuación describiremos el método PT el cuál es del tipo PC y fue recientemente propuesto en [Martín, 2014]. Para esto, consideremos la función:

$$\tilde{F}(x, \alpha) = \begin{pmatrix} \sum_{i=1}^k \alpha_i \nabla f_i(x) \\ \sum_{i=1}^k \alpha_i - 1 \end{pmatrix} = 0. \quad (3.22)$$

La idea del PT es separar los espacios  $x$ (decisión) y  $\alpha$ (pesos) cuando sea posible.

#### Predictor:

Se desea obtener el núcleo de  $\tilde{F}'(x, \alpha)$ . Para esto, sea  $x$  un punto de KKT,  $v \in \mathbb{R}^n$  y  $\mu \in \mathbb{R}^k$  sujeto a

$$\tilde{F}'(x, \alpha) = \begin{pmatrix} \sum_{i=1}^k \alpha_i \nabla^2 f_i(x) & \nabla f_1(x) & \dots & \nabla f_k(x) \\ 0 & 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} \nu \\ \mu \end{pmatrix} = 0 \quad (3.23)$$

Definimos  $W_\alpha := \sum_{i=1}^k \alpha_i \nabla^2 f_i(x)$  una matriz regular y

$$J = J(x) = \begin{pmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_k(x)^T \end{pmatrix}.$$

Supongamos  $\mu \neq 0$  y que  $W_\alpha$  es una matriz regular, que satisface

$$\sum_{i=1}^k \mu_i = 0. \quad (3.24)$$

De la primera ecuación de (3.23) tenemos:

$$\nu_\mu = -W_\alpha^{-1} J^T \mu. \quad (3.25)$$

Luego, dada una dirección  $d \in \mathbb{R}^k$ , tal que  $J\nu_\mu = d$ , podemos encontrar  $\mu_d$  que resuelva el siguiente sistema

$$\begin{pmatrix} -JW_\alpha^{-1} J^T \\ 1 \dots 1 \end{pmatrix} \mu_d = \begin{pmatrix} d \\ 0 \end{pmatrix} \quad (3.26)$$

Para obtener las direcciones  $d$ , aplicamos una descomposición del tipo QR a  $\alpha$

$$\alpha = QR := (q_1, \dots, q_k)R,$$

lo que implica que

$$d_i = q_{i+1} \quad i = 1, \dots, k-1. \quad (3.27)$$

Con esto, los movimientos  $J\nu_i$  forman una base ortonormal del FP linearizado. Ya con  $\mu_d$  calculado, se obtiene  $\nu_{\mu_d}$  como en (3.25). Faltaría aplicar el predictor de la siguiente manera:

$$p := x + t\nu_\mu. \quad (3.28)$$

Como estamos interesados en una distribución uniforme de puntos en el FP, se asume que cada objetivo es Lipschitz continuo, entonces queremos que para dos soluciones consecutivas  $x_i$  y  $x_{i+1}$  se cumpla,

$$\|F(x_i) - F(x_{i+1})\|_2 \approx \tau,$$

donde  $\tau > 0$ . De este modo, un tamaño de paso adecuado está dado por

$$t = \frac{\tau}{\|J\nu_\mu\|_2}. \quad (3.29)$$

### Corrector

Para el corrector se aplica el método de Newton Multiobjetivo [Fliege et al., 2009]. La dirección de Newton está definida como la solución de

$$\begin{array}{ll} \min_{(\nu, \delta) \in \mathbb{R}^n \times \mathbb{R}} & \delta \\ \text{s.a} & \nabla f_i(x)^T \nu + \frac{1}{2} \nu^T \nabla^2 f_i(x) \nu \leq \delta, \quad i = 1, \dots, k. \end{array} \quad (3.30)$$

### 3.2.3 Discusión

Una de las ventajas del método radica en la separación de espacios, esto debido a que no aumenta la no linealidad del conjunto solución a diferencia de otros métodos PC (ver por ejemplo [Hillermeier, 2001]). Debido a que se controla el tamaño de paso, es posible obtener una buena distribución de las soluciones obtenidas.

Por otro lado, una desventaja es el uso de información de segundo orden para llevar a cabo el procedimiento corrector. En este trabajo como nos centramos en el caso biobjetivo el corrector no es utilizado debido a que nuestro CP es de dimensión uno. Otra desventaja es la que se presenta al trabajar con funciones con FP desconectados, ya que sólo se puede generar una de las componentes de él. Estas observaciones se muestran en la sección de ejemplos.

**Ejemplo 3.6.** (Función de Dent) Consideremos el siguiente POM:

$$\text{Minimizar } F = \left[ \begin{array}{l} \frac{1}{2} \left( \sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} + x_1 - x_2 \right) + \lambda \exp - (x_1 - x_2)^2 \\ \frac{1}{2} \left( \sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} - x_1 + x_2 \right) + \lambda \exp - (x_1 - x_2)^2 \end{array} \right]$$

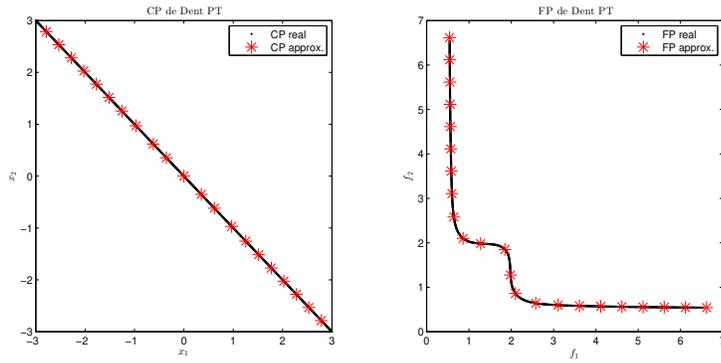
donde  $\lambda = 0.85$ . Esta función tiene un FP conectado, con regiones tanto convexas como no convexas. En la Figura 3.9 se puede observar el FP y CP reales con . color negro y el FP y el CP obtenidos al aplicar el método con \* color rojo. Podemos apreciar que las soluciones obtenidas cubren uniformemente el FP de la función, además de poseer una buena extensión.

**Ejemplo 3.7.** (Función de Fonseca) Recordemos que esta función tiene un FP no convexo y un CP conectado. En la Figura 3.10 se puede observar el FP y CP reales con . color negro y el FP y el CP obtenidos al aplicar el método con \* color rojo. Observando los resultados obtenidos, vemos que al igual que en el ejemplo anterior se obtuvo una buena solución dado que se cubre todo el FP y CP además de que se cuenta con una distribución uniforme de soluciones óptimas.

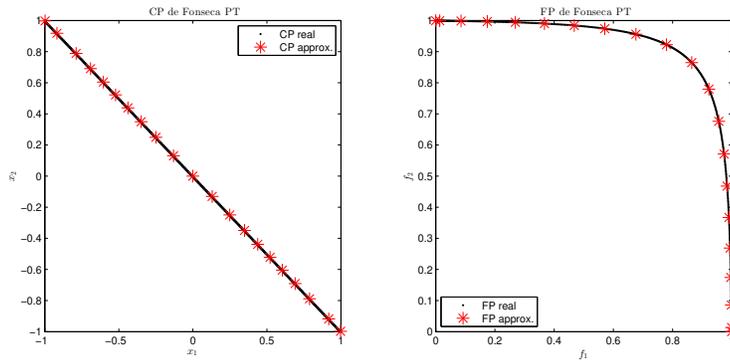
**Ejemplo 3.8.** (Función de Murata) consideremos el siguiente POM

$$\text{Minimizar } F = \left[ \begin{array}{l} 2\sqrt{x_1} \\ x_1(1 - x_2) + 5 \end{array} \right]$$

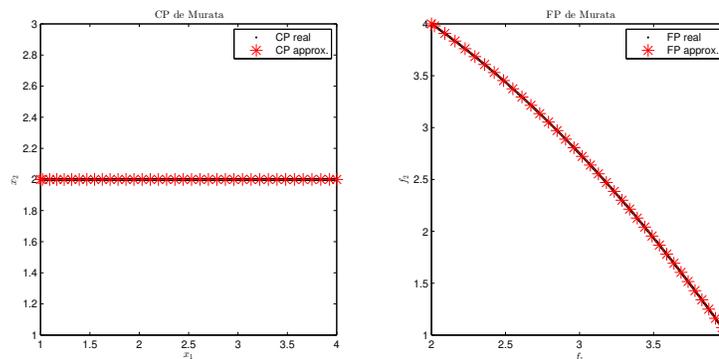
donde  $1 \leq x_1 \leq 4$  y  $1 \leq x_2 \leq 2$ . Esta función tiene un FP no convexo y su CP conectado. En la Figura 3.11 puede observar el FP y CP reales con . color negro y el FP y el CP obtenidos al aplicar el método con \* color rojo. En este ejemplo se observan buenos resultados al obtener tanto una buena extensión del FP y del CP, como una buena distribución de soluciones a su vez.



**Figura 3.9:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función de Dent.



**Figura 3.10:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado mediante PT en color rojo. Ambos resultados son de la función de Fonseca.



**Figura 3.11:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Murata.

**Ejemplo 3.9.** (Función ZDT1) Este POM posee un FP convexo y ya ha sido previamente definido en la sección anterior. Hace uso de 30 variables. En la Figura 3.12 puede observar el FP real con . color negro y el FP obtenido al aplicar el método con \* color rojo. Podemos observar un buen resultado, debido a que se construye todo el FP y con una buena distribución.

**Ejemplo 3.10.** (Función ZDT3) Al igual que el caso anterior, se trata de un POM que considera 30 variables de decisión. Esta definido como:

$$\text{Minimizar } F = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix}$$

donde

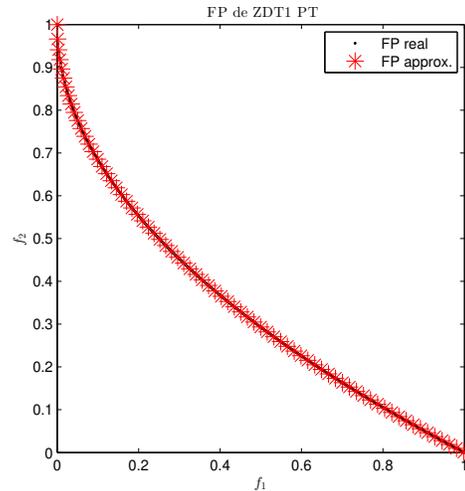
$$f_1(x) = x_1$$

$$f_2(x, g) = g(x) \left( 1 - \sqrt{\frac{f_1}{g(x)}} - \frac{f_1}{g(x)} \sin(10\pi f_1) \right)$$

y

$$g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^{30} x_i$$

con  $0 \leq x_i \leq 1$  para  $i = 1, \dots, 30$ . Esta función posee un FP convexo y no conectado. En la Figura 3.13 se puede observar el FP real con . color negro y el FP obtenido al aplicar el método con \* color rojo. Note que con este método solo se logra obtener una componente del FP, lo que sería una desventaja si

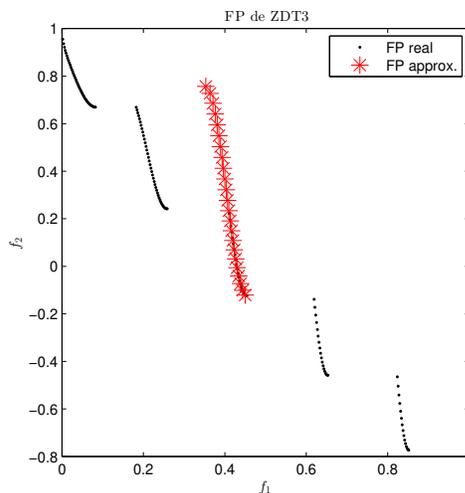


**Figura 3.12:** En la imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. El resultado es de la función ZDT1.

lo que se busca es construir todo el FP. Por otro lado vemos que el resultado obtenido con el PT genera soluciones que no pertenecen al FP real, esto porque no hay manera de que el método revise que son puntos dominados por otros pertenecientes a las demás componentes del FP real.

### 3.2.4 Experimentos numéricos

En esta sección se muestra el resumen de los resultados obtenidos en los experimentos numéricos que se desarrollaron en este trabajo para el método de PT. Las funciones de prueba utilizadas están descritas en el apéndice 6. La tabla 3.2 muestra los costos en términos de evaluaciones de la función requeridos para generar todo el FP (EV.F), el número de evaluaciones de la función promedio requeridas para generar un punto (EV.F/P) y tiempo del método, para cada problema de prueba, considerando su éxito mediante la medida  $\Delta_2$ . Para cada prueba se generaron 20 puntos. Se considera que los problemas marcados en negro, en la tabla, fueron resueltos exitosamente pues los valores del indicador  $\Delta_2$  son menores a 0.01 además de la comprobación visual de los conjuntos y frentes de Pareto. Observe que los peores valores de  $\Delta_2$  se obtuvieron en *Poloni*, *Schaffer 2*, **ZDT3** que son funciones cuyo FP es desconectado, de modo que solo se genera una de las componentes de dicho frente. Por otro lado el valor de  $\Delta_2$  en *Dent* es muy elevado también, esto se debe a que el PT genera todos los puntos de KKT de la función muchos de los cuales no pertenecen al FP real. En *Binh*, *Laumanns*, *Lis* note que aunque no se registran como exitosos el



**Figura 3.13:** En la imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. El resultado es de la función ZDT3.

valor del indicador es bueno, pues ya es cercano a cero. Podemos concluir que el PT devuelve soluciones bien distribuidas y logra buenas aproximaciones del FP siempre y cuando este no sea desconectado. También note que su costo en cuanto a evaluaciones de la función por punto no se eleva de manera radical aunque requiere información de primer y segundo orden. Es importante mencionar que el punto inicial que recibe el PT ya forma parte del FP real, el costo de evaluaciones de la función presenta un aumento si el punto inicial dado no pertenece al FP real. Observe que el PT funciona adecuadamente en  $ZDT_4$ , esto se debe a que el punto inicial que recibe ya forma parte del FP real, por lo que de entrada se descarta un estancamiento en los frentes locales. Esto no ocurre en el método híbrido del cual se habla en capítulos posteriores.

**Tabla 3.2:** Resultados numéricos del Método Pareto Tracer.

PARETO TRACER				
Función	# EV.F	# EV.F/P	Tiempo(segs.)	$\Delta_2$
Binh	860	43	9.2569	0.0443
Dent	860	43	2.1408	1.0773
Fonseca	860	43	14.5899	<b>0.0013</b>
Laumanns	860	43	3.1572	0.0109
Lis	860	43	4.2197	0.0139
Murata	860	43	2.1433	<b>0.0018</b>
Poloni	860	43	15.12484	0.9033
Rendon 2	860	43	2.2806	<b>0.0075</b>
Schaffer 1	420	21	2.3727	<b>0.0080</b>
Schaffer 2	420	21	3.6157	0.3015
ZDT1	78, 140	3907	2.5149	<b>0.0011</b>
ZDT2	78, 140	3907	3.3649	<b>0.0010</b>
ZDT3	78, 140	3907	9.7993	0.2780
ZDT4	10, 140	507	7.9042	<b>0.0017</b>



## Capítulo 4

# Método poblacional

### 4.1 Introducción

Las heurísticas son métodos diseñados *a modo* para resolver problemas complejos. Por ejemplo, aquellos problemas cuyo espacio de búsqueda es muy grande o donde los espacios o las funciones no cumplen con las hipótesis requeridas por los métodos convencionales (convexidad, diferenciabilidad, etc.). Estos métodos son muy útiles cuando se busca resolver un problema real en el cual se quieren respetar las peculiaridades que éste posee, pues da margen a modelaciones menos rígidas. La computación evolutiva [Quintero and Coello, 2006] es un área de investigación de las ciencias de la computación que se ha empleado con éxito para aproximar soluciones de problemas de optimización. Se inspira en los procesos de la evolución de las especies, observados en la naturaleza [Coello and Zacatenco, 2004]. El proceso de evolución natural<sup>1</sup> se considera en un ambiente compuesto por una población de individuos que luchan por sobrevivir y reproducirse. La aptitud de cada uno de los individuos, se relaciona con el hecho de qué tan “bien” cumplen sus metas, lo que representa sus oportunidades de sobrevivir y multiplicarse. Desde el punto de vista del proceso de solución de problemas, dentro de este paradigma se consideran una colección de posibles soluciones del problema (individuos), y la calidad de dichas soluciones (aptitud). La calidad de las soluciones se refiere a que tan bien resuelven el problema que se está tratando. La calidad determina la oportunidad que tiene cada posible solución para permanecer dentro de la población y de ser utilizados como semillas para la construcción de nuevos candidatos; de aquí que se considere como un método iterativo y *poblacional*. Los algoritmos poblacionales son aquellos métodos numéricos cuya principal característica radica en que se trabaja, en cada

---

<sup>1</sup>El termino “evolución” se deriva del verbo en latín *evolvere*, describiendo un proceso de despliegue.

iteración, con una población de posibles soluciones a un cierto problema. Esta forma de operar los hace particularmente atractivos para resolver POMs. Esto es porque, en contraposición con los algoritmos de programación matemática, que son puntuales, *i.e.*, donde se trabaja con un único punto inicial y se obtiene una solución a la vez, los métodos poblacionales arrojan todo un conjunto de soluciones aproximadas en una sola ejecución.

## 4.2 Preliminares

**Definición 4.1.** *Un cromosoma es una estructura de datos que contiene una cadena particular de parámetros de diseño.*

Ejemplos de cromosomas de longitud  $n$  pueden ser una cadena de bits  $c \in \mathbb{Z}_2^n$ , un arreglo de números enteros que representan a cierto  $c \in \mathbb{Z}^n$ , una cadena de caracteres, un vector de números de punto flotante  $c \in \mathbb{R}^n$ , etc.

**Definición 4.2.** *Un gen es una subsección de un cromosoma que (usualmente) codifica el valor de un solo parámetro.*

**Definición 4.3.** *Un individuo es la representación de una solución al problema. Cada individuo está en correspondencia con un cromosoma particular.*

**Definición 4.4.** *Llamaremos aptitud al valor que se asigna a cada individuo y que indica que tan bueno es éste con respecto a los demás para la solución de un problema.*

**Definición 4.5.** *Una población es un conjunto finito de individuos.*

**Definición 4.6.** *Llamaremos generación<sup>2</sup> al proceso iterativo mediante el cuál se crea una nueva población a partir de la existente mediante ciertos operadores de variación.*

**Definición 4.7.** *Un operador de variación es cualquier mecanismo que influye la forma en que se pasa la información genética de una generación a otra.*

- a) *Se conoce como cruce al operador de variación que forma un nuevo cromosoma combinando partes dos cromosomas padres.*
- b) *Se conoce como mutación al operador de variación que forma un nuevo cromosoma a través de alteraciones de los valores de los genes de un cromosoma padre.*

---

<sup>2</sup>El número de generaciones se puede utilizar como criterio de terminación de este tipo de programas; también se puede emplear como punto de partida para realizar comparaciones entre algoritmos.

c) Se conoce como *reordenamiento al operador de variación* que cambia el orden de los genes de un cromosoma, con la esperanza de juntar los genes que se encuentran relacionados.

Los *algoritmos evolutivos* son un conjunto de técnicas numéricas que suponen que dada una población de individuos, la presión de supervivencia del entorno, ocasionada por la selección natural causa un aumento en la aptitud promedio de dicha población. Dado un cierto criterio por optimizar, éste puede adecuarse manera de función de correspondencia de aptitud para una cierta solución. Entonces, se puede generar aleatoriamente un conjunto finito de posibles soluciones, es decir, elementos del dominio de la función a los que se les aplica una adaptación de la función objetivo como medida de la aptitud. Basándonos en la aptitud, algunos de los mejores individuos son candidatos para ser semillas de la siguiente generación al aplicares los operadores de variación. Los nuevos individuos, creados a partir de los mejores, compiten con los anteriores por un lugar en la siguiente generación; este proceso se puede iterar hasta que se encuentra un candidato con la calidad suficiente para ser solución de nuestro problema. Cabe mencionar que durante la selección, los individuos más aptos tienen una mayor oportunidad de ser elegidos que los menos aptos, pero todo individuo tiene una cierta, mayor o menor, probabilidad de reproducción. Similarmente, para la mutación, la porción que va a ser mutada del genotipo de un candidato y la nueva porción que va a ser el reemplazo son elegidos aleatoriamente. Existen dos tipos de selección: la selección de reproducción y la selección de supervivencia ó reemplazo. El Algoritmo 1 describe el proceso general de la computación evolutiva. La idea general es generar de manera aleatoria una población de soluciones factibles, después se evalúa la aptitud de cada uno de ellos (recordemos que la aptitud está definida en términos del valor de la función objetivo evaluada en el punto en cuestión). En cada iteración se selecciona a los padres, a los cuales se les aplican los operadores de variación; posteriormente se evalúan los nuevos candidatos obtenidos y de la colección de todos los individuos se seleccionan los que pasarán a la siguiente generación. Para la terminación del proceso se establecen diversos criterios de paro.

Desde los años cuarenta surgió la idea de aplicar los principios darwinianos para resolver problemas. A partir de los años sesenta se empezaron a desarrollar diferentes implementaciones de la idea básica. En E.U.A, Gary B. Fogel, A.J Owens y M.J Walsh presentaron la **programación evolutiva**, mientras que John Holland llamó a su método **algoritmo genético**. En Alemania, Ingo Rechenberg y Hans-Paul Schwefel inventaron las **estrategias evolutivas**[Eiben, 1998]. A continuación describiremos brevemente cada uno de estos paradigmas.

La **programación evolutiva** fue uno de los primeros intentos por simular la co-evolución <sup>3</sup>[Fogel, 1965][Fogel, 1966]. La implementación de los **algorit-**

---

<sup>3</sup>La co-evolución se define como el cambio evolutivo que acontece en especies interactuantes y que está mediado por la adaptación evolutiva.

**mos genéticos** involucra el estudio de los procesos lógicos involucrados en la adaptación [Holland, 1962b]. Un algoritmo genético es un sistema adaptativo general, donde los programas cuentan con una población que interactúa y cuyas soluciones mejoran con base en el ambiente que determina lo apropiado de su comportamiento, al combinar variaciones aleatorias con un proceso de selección. La evolución se da a nivel de los individuos (genotipos), su operador principal es la cruce. Se aplica en problemas de optimización, bases de datos, reconocimiento de patrones, planeación de movimientos de robot, entre otros [Holland, 1962a]. Las **Estrategias Evolutivas** se originaron cuando Rechenberg y Schwefel estudiaban la mecánica de fluidos con énfasis en la experimentación con un túnel de viento. Este estudio buscaba optimizar la forma de un tubo curvo, la minimización del arrastre de la placa de unión y la optimización estructural de la boquilla intermitente de dos fases [Rechenberg, 1973]. Dichos problemas no se podían resolver analíticamente, por lo que Rechenberg decidió desarrollar un método de ajustes discretos aleatorios inspirado en el mecanismo de mutación que ocurre en la naturaleza. Rechenberg decidió efectuar dichos cambios siguiendo una distribución binomial de varianza fija, teniendo como base que en la naturaleza las mutaciones pequeñas ocurren con mayor frecuencia que las grandes [Bäck, 1996]. Existen diferentes áreas de estudio dentro de los algoritmos evolutivos, las ciencias de la computación se enfocan en incrementar su eficiencia y construir operadores de variación específicos para generar soluciones mejores o de manera más eficiente. En este capítulo estudiamos un tipo de algoritmo genético particular que es adaptado para la aproximación de soluciones en el caso multiobjetivo.

---

**Algoritmo 1** Algoritmo Evolutivo

---

- 1: **Inicializar** *población* con soluciones posibles.
  - 2: **Evaluar** cada solución.
  - 3: **Repetir** paso 4 a paso 8 **hasta** que el criterio de paro se cumpla
  - 4: **Seleccionar** padres.
  - 5: **Recombinar** parejas de padres.
  - 6: **Mutar** los hijos resultantes.
  - 7: **Evaluar** a los nuevos candidatos.
  - 8: **Seleccionar** los individuos que pasarán a la siguiente generación.
  - 9: Fin
- 

### 4.3 Algoritmo NSGA-II

El primer *Algoritmo Genético* (AG) fue propuesto por Holland en 1975. Los AGs imitan los principios básicos de la selección natural formulados por Charles Darwin [Darwin, 1929] en 1859 y Gregory Mendel en 1865 [Mendel, 1865]. Se basan en tres principios básicos:

1. Existe una población de soluciones posibles al problema.
2. Los operadores de variación crean nuevas soluciones con propiedades similares a las ya existentes. Los operadores de variación de los *Algoritmos Genéticos* (AGs) son la cruce y la mutación.
3. Los individuos más aptos tienen mayor probabilidad de ser seleccionados para reproducirse.

En el Algoritmo 2 se describe el AG simple. Utiliza una población constante  $P$  de tamaño  $N$  fijo durante todo el proceso. Usualmente los individuos de  $P$  están codificados en binario, en cadenas de longitud  $l$ . Los operadores de variación actúan directamente sobre esta codificación. Actualmente, codificar a los individuos en binario o no, depende de la naturaleza de las soluciones del problema, y el programador decide en qué caso es conveniente.

---

**Algoritmo 2** Algoritmo Genético.
 

---

```

1: Se crea una población inicial  $P = \{x^1, \dots, x^n\}$ ,  $|P| = N$  de posibles soluciones.
2: for  $i = 1 : N$  do  $f(x^i)$ 
3: end for
4: while Criterio de terminación no se cumpla do
5:    $M = \{ \}$ 
6:   Insertar  $N$  individuos de  $P$  en  $M$ .
7:   Revolver a los individuos de  $M$ .
8:    $ind = 1$ 
9:   repeat
10:    if  $rand(0,1) \leq Prob_{cruza}$  then Recombina  $x^{ind}$  y  $x^{ind+1}$  y colocar hijos en  $P'$ .
11:    else
12:      Copiar  $x^{ind}$  y  $x^{ind+1}$  en  $P'$ .
13:    end if
14:     $ind = ind + 2$ 
15:  until  $ind > N$ 
16:  for  $i = 1 : N$  do
17:    for  $j = 1 : l$  do
18:      if  $rand(0,1) \leq Prob_{mut}$  then Mutar  $(x_j^i)$ , donde  $x_j^i \in P'$ .
19:    end if
20:  end for
21:  Calcular  $f(x^i)$  con  $x^i \in P$ .
22: end for
23:  $P = P'$ 
24: end while

```

---

Los AGs se pueden extender al caso multiobjetivo, los *Algoritmos Genéticos Multiobjetivo* (AGMOs). El uso de AGMOs para resolver POMs presenta

varias ventajas. Una de ellas, es que al ser algoritmos poblacionales se obtiene todo un conjunto de soluciones en una sola ejecución. Así mismo, este tipo de métodos no requiere que las funciones objetivo cumplan ciertos requerimientos para poder aplicarse por ejemplo el ser diferenciables. En tiempos recientes los AGMOs son cada vez más utilizados para resolver problemas reales [Toscano Pulido, 2005]. Por ejemplo se han utilizado exitosamente en programación de horarios de trabajo de máquinas [Murata et al., 1996], en minería de datos [Ishibuchi and Yamamoto, 2004] y para diseñar una superficie de sustentación y un compuesto de cerámica laminada [Belegundu et al., 1994].

En el *Algoritmo Genético Multiobjetivo* (AGMO) la selección de supervivencia de individuos se realiza de acuerdo a la relación de dominancia de Pareto. La meta es obtener, a partir de la población, una aproximación razonable del FP real del problema.

El NSGA-II es un AGMO propuesto por K. Deb en [Deb et al., 2000], el cual se ha vuelto muy popular ya que es un algoritmo eficiente y eficaz<sup>4</sup>. El NSGA-II emplea una *jerarquización de Pareto* e incluye un mecanismo para preservar la diversidad de la población. Además, incorpora el *elitismo*, es decir, retiene a las mejores soluciones generadas durante la búsqueda. Esto se logra combinando en cada generación la población de padres con la población de hijos, logrando así que compitan entre sí para pasar a la siguiente generación. En lo posterior se describe el funcionamiento particular del algoritmo.

### **Jerarquización de Pareto:**

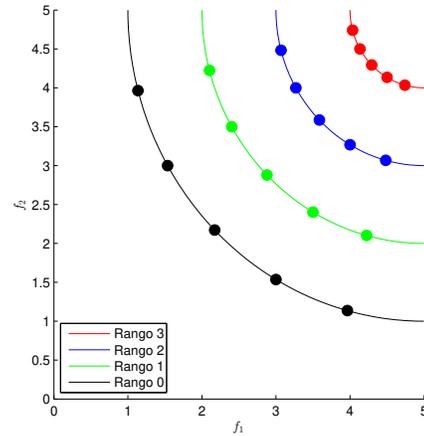
La jerarquización de Pareto propuesta en el NSGA-II consiste en clasificar los individuos en varias categorías. Para realizar dicha clasificación se llevan a cabo los siguientes pasos:

1. El conjunto de individuos no dominados pertenece al primer nivel en la jerarquización de Pareto. Llamado *rango*<sub>0</sub>.
2. Se elimina los individuos *rango*<sub>0</sub> de la población y se encuentra el nuevo conjunto de individuos no dominados en la población restante. Este conjunto constituye el segundo nivel en la jerarquía, llamado *rango*<sub>1</sub>.
3. El proceso continúa hasta que todos los individuos hayan sido clasificados, es decir, todos cuenten con un rango asignado.

Observamos que si  $m$  es el número de objetivos y  $N$  el número de individuos en la población entonces el procedimiento descrito anteriormente requiere  $O(mN)$  comparaciones para decidir si un individuo se encuentra en el primer nivel de la jerarquía. Para completar el primer nivel se habrán utilizado  $O(mN^2)$

---

<sup>4</sup>A la fecha, el algoritmo NSGA-II reúne más de 16,400 citas en revistas de investigación y aplicaciones



**Figura 4.1:** Ejemplo de una clasificación en jerarquías de Pareto en un problema con dos objetivos. Los puntos son las soluciones generadas hasta el momento.

comparaciones. Si en cada nivel existen  $O(N)$  individuos, entonces para realizar toda la clasificación se requieren  $O(mN^3)$  comparaciones. Actualmente existe una variación en este proceso que reduce el número de comparaciones a  $O(mN^2)$ .

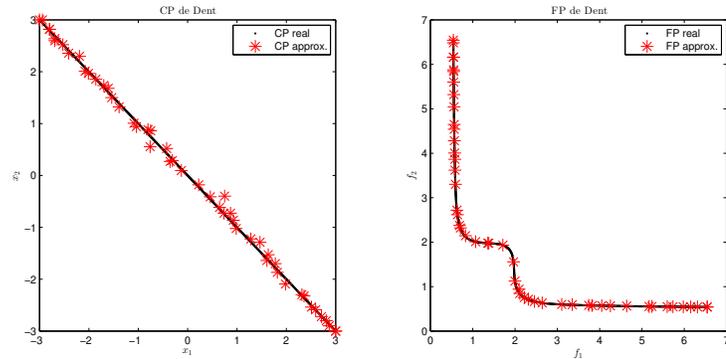
#### Preservación de la diversidad:

Para preservar la diversidad en la población, el NSGA-II cuenta con un estimador de densidad simple y efectivo. La meta de este indicador, es ordenar los individuos que tengan un mismo rango con respecto a la jerarquización de Pareto de modo que los individuos que se encuentren en regiones menos pobladas del espacio de búsqueda sean favorecidos por el mecanismo de selección.

La densidad se estima calculando la distancia promedio de los dos individuos a cada lado de una solución para cada uno de los objetivos. Luego, un individuo con una medida mayor se encuentra en una región menos poblada que un individuo con menor medida.

#### Selección de individuos

La característica más destacable del NSGA-II radica en la forma en que se realiza la selección: al comparar dos individuos se elige aquel que pertenezca a la jerarquía de Pareto más baja. En caso de que ambos individuos pertenezcan a la misma categoría, entonces se elige a aquél individuo ubicado en una región con menor densidad de población. Dado que la selección, se aplica a la unión de padres e hijos, es implícitamente elitista, siempre se conservan los individuos que tengan un mejor rango, independientemente de si pertenecen a los padres o a los hijos.

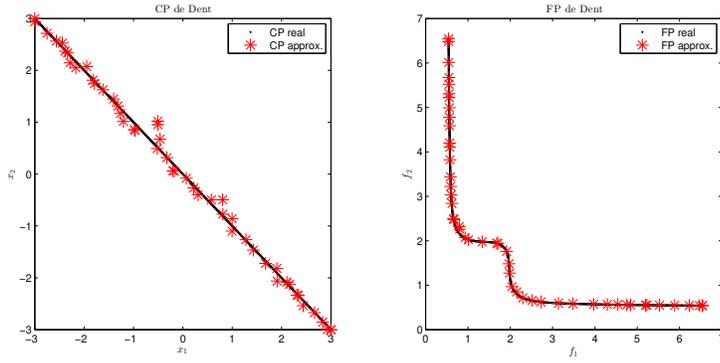


**Figura 4.2:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Dent para 600 generaciones.

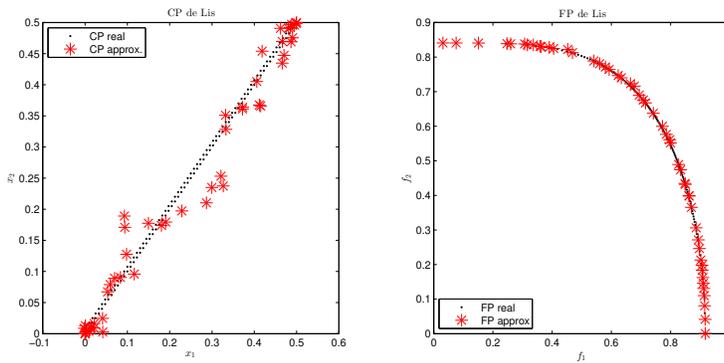
A continuación se muestran los resultados obtenidos al aplicar el NSGA-II a ciertas funciones de prueba, algunas ya fueron descritas en secciones anteriores. Cada subsección contiene la comparación de los CP y FP obtenidos con el método contra los CP y FP teóricos. Los resultados presentados en los ejemplos se obtuvieron al aplicar el proceso durante 600 generaciones utilizando 30,000 evaluaciones de la función objetivo y durante 1200 generaciones utilizando 60,000 evaluaciones de la función.

**Ejemplo 4.1.** (Función de Dent) Recordemos que este POM tiene un FP con regiones convexas y no convexas. Su CP es conectado. En la Figura 4.2 se puede observar el FP y CP reales con  $\cdot$  color negro y el FP y el CP obtenidos al aplicar el método con  $*$  color rojo. Podemos observar que con este número limitado de recursos 30,000 evaluaciones de la función ya se obtiene casi todo el FP y el CP de la función, pero no se cuenta con una distribución uniforme de los puntos obtenidos. Sin embargo, como se verá en la sección 4.4 si el algoritmo se deja iterar más tiempo, se logra una mejor distribución. Con el consabido aumento en el número de evaluaciones de la función objetivo. Esto se puede apreciar en la Figura 4.3.

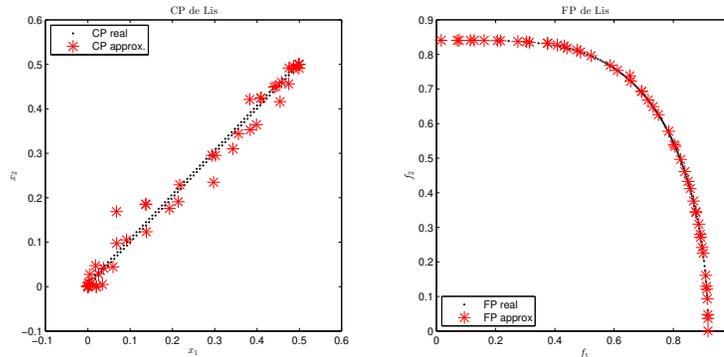
**Ejemplo 4.2.** (Función de Lis) Recordemos que este POM tiene un FP no convexo y no conectado. Su CP es no conectado. En la Figura 4.4 se puede observar el FP y CP reales con  $\cdot$  color negro y el FP y el CP obtenidos al aplicar el método con  $*$  color rojo. Podemos observar que se obtiene una buena aproximación de todo el FP. Mientras que el CP aproximado, cuenta con puntos no tan próximos al CP real. Igualmente no se cuenta con una distribución uniforme de los puntos obtenidos. En la Figura 4.5 se observa que la distribución de puntos va mejorando conforme aumentan los recursos computacionales, pero se sigue sin conseguir una buena aproximación al CP real.



**Figura 4.3:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Dent para 1200 generaciones.



**Figura 4.4:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Lis para 600 generaciones.

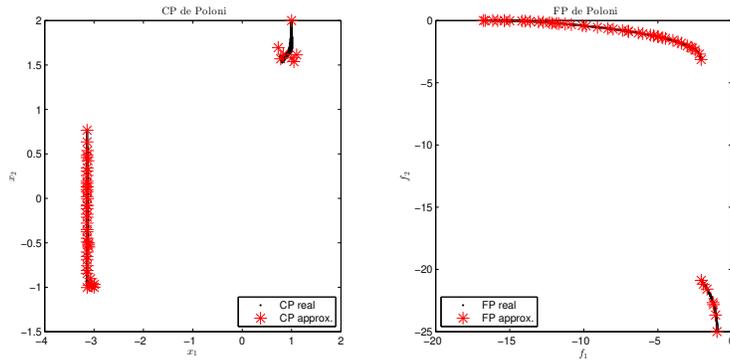


**Figura 4.5:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Lis para 1200 generaciones.

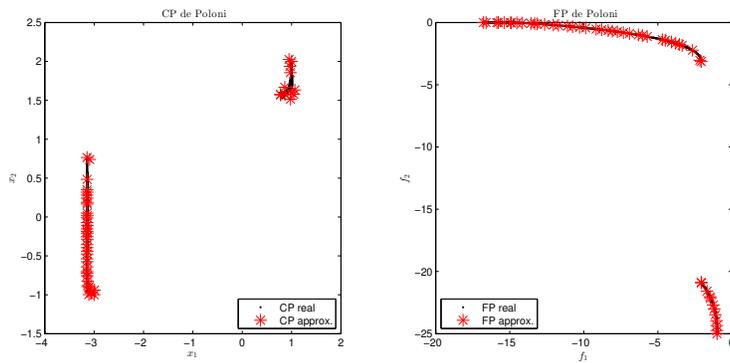
**Ejemplo 4.3.** (Función de Poloni) Recordemos que se trata de un problema de maximización. El CP es no conectado al igual que el FP y ninguno parece pertenecer a una función bien definida, esto debido a las restricciones que se manejan y a las combinaciones de funciones trigonométricas. Por otro lado aunque el FP no es convexo. En la Figura 4.6 se puede observar el FP y CP reales con  $\cdot$  color negro y el FP y el CP obtenidos al aplicar el método con  $*$  color rojo para 600 generaciones. Note que tanto en el FP y el CP obtenidos por el método tienen buena extensión. De manera similar, no se cuenta con una distribución uniforme de aproximaciones. Observe que aunque el FP y el CP son no conectados, no se obtienen soluciones que no pertenezcan a las componentes reales. Note que a diferencia del SP o PT, los resultados obtenidos con el NSGA-II son mejores en cuanto a extensión y distribución de puntos. En este ejemplo al aumentar los recursos computacionales las mejoras ya no son significativas, de modo que con 30,000 evaluaciones de la función basta para obtener una buena aproximación.

**Ejemplo 4.4.** (Función de Schaffer 2) Esta función tiene un FP y un CP desconectado. En la Figura 4.8 se puede observar el FP y CP reales con  $\cdot$  color negro y el FP y el CP obtenidos al aplicar el método con  $*$  color rojo. Note que en esta función se obtienen buenos resultados, pues se obtienen una buena aproximación del FP y del CP reales. Análogamente al ejemplo de Poloni, al aumentar el número de evaluaciones de la función no modifica los resultados de manera notoria.

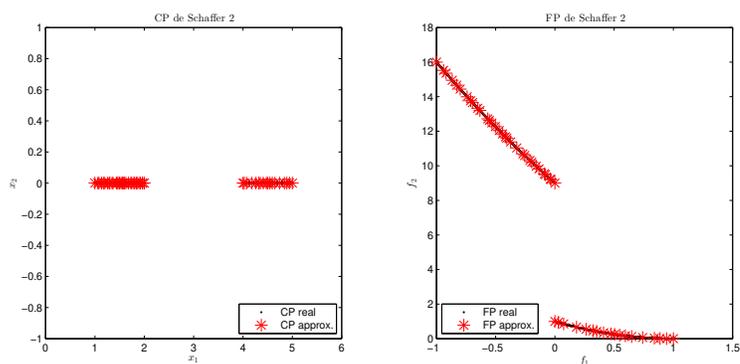
**Ejemplo 4.5.** (Función ZDT3) Este POM posee un FP convexo y no conectado. Se consideran 30 variables. En la Figura 4.10 se puede observar el FP real con  $\cdot$  color negro y el FP obtenido al aplicar el método con  $*$  color rojo. En la primera imagen podemos observar que con este número limitado de recursos se va aproximando el FP de una manera un tanto uniforme, pero no se llega al FP



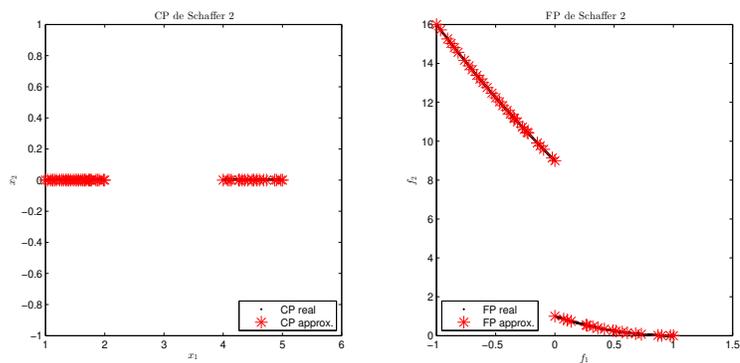
**Figura 4.6:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Poloni para 600 generaciones.



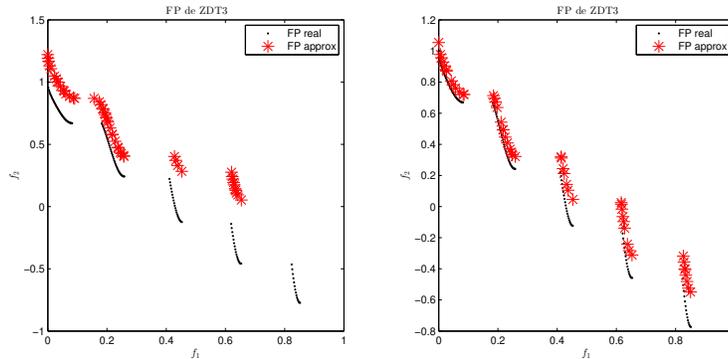
**Figura 4.7:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Poloni para 1200 generaciones.



**Figura 4.8:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Schaffer 2 para 600 generaciones.



**Figura 4.9:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Schaffer 2 para 1200 generaciones.



**Figura 4.10:** En la imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función ZDT3 para 600 generaciones y 1200 generaciones respectivamente.

real. Se recomienda aumentar el número de generaciones para que la solución converja al FP. En la segunda imagen se puede observar que con el doble de evaluaciones de la función la aproximación al FP mejora, se van consiguiendo todas las componentes y además se va teniendo una buena distribución de soluciones. Una de las grandes ventajas del NSGA-II es poder generar todas las componentes de las funciones con frentes desconectados, como sucede también en Poloni y Schaffer 2.

## 4.4 Resultados numéricos

En esta sección se muestra el resumen de los resultados obtenidos en los experimentos numéricos que se desarrollaron en este trabajo para el método de NSGA-II. Las funciones de prueba utilizadas están descritas en el apéndice 6. La tabla 4.1 muestra los costos en términos de evaluaciones de la función requeridos para generar todo el FP (EV.F), el número de evaluaciones de la función promedio requeridas para generar un punto (EV.F/P) y tiempo del método, para cada problema de prueba, considerando su éxito mediante la medida promedio  $\Delta_2$ . Para cada prueba se trabajó con una población de 50 individuos, se ejecutó 30 veces el algoritmo para 1200 generaciones lo que equivale a 60,000 evaluaciones de la función objetivo. Se considera que los problemas marcados en negro, en la tabla, fueron resueltos exitosamente pues los valores del indicador  $\Delta_2$  son menores a 0.01 además de la comprobación visual de los conjuntos y frentes de Pareto. Note que con este método la mayoría de los casos (11 de 14 funciones de prueba) cumplieron el criterio impuesto para considerarse resultados exitosos, lo que nos habla de un método eficaz pues cumple con devolver una buena aproximación del FP y del CP. La desventaja principal radica en el

Tabla 4.1: Resultados numéricos del NSGA-II.

NSGA-II				
Función	# EV. F	# EV. F/P	Tiempo(segs.)	$\Delta_2$
Binh	60,000	1,200	0.2583	0.0271
Dent	60,000	1,200	0.3663	<b>0.0030</b>
Fonseca	60,000	1,200	14.8089	<b>0.0026</b>
Laumanns	60,000	1,200	22.6632	<b>0.0068</b>
Lis	60,000	1,200	16.8759	<b>0.0079</b>
Murata	60,000	1,200	14.8109	<b>0.0010</b>
Poloni	60,000	1,200	15.0285	<b>0.0053</b>
Rendon 2	60,000	1,200	13.5660	<b>0.0044</b>
Schaffer 1	60,000	1,200	14.3617	<b>0.0039</b>
Schaffer 2	60,000	1,200	14.3828	<b>0.0020</b>
ZDT1	60,000	1,200	16.9936	<b>0.0057</b>
ZDT2	60,000	1,200	24.4992	0.0103
ZDT3	60,000	1,200	21.6047	<b>0.0089</b>
ZDT4	60,000	1,200	8.2797	0.6422

costo de evaluaciones de la función que es elevado, esto lo vuelve un método no muy eficiente comparando con los resultados de los capítulos anteriores. Por otro lado vemos que en la función  $ZDT_4$  el valor de  $\Delta_2$  es malo, esto es porque el método poblacional se estanca en un FP local de la función. Recordando los métodos anteriores en esta función se obtuvieron buenos resultados. Por ejemplo el SP garantiza que los puntos generados son óptimos de Pareto por lo que no se tiene el problema de encontrarse en un frente local. Con el NBI tampoco existe esta situación, puesto que el punto que se obtiene con este método pertenece a la frontera de la imagen justamente donde se localiza el frente global de dicha función. Y en el caso del PT aunque es de tipo local y podría ocurrir lo mismo, esto se evita al darle una solución inicial que se encuentre en el FP global.

# Capítulo 5

## Método Híbrido.

### 5.1 Análisis preliminar

Los métodos híbridos que combinan los algoritmos evolutivos con algún algoritmo de búsqueda local, como por ejemplo un método basado en gradientes, tienen como meta mejorar el desempeño del algoritmo básico. En este capítulo, se presenta un método híbrido original que es propuesto con el fin de aprovechar las ventajas poblacionales que presenta el método NSGA-II mientras se explotan las cualidades observadas del método PT. En general se han clasificado [Sinha and Goldberg, 2003] tres tipos de métodos híbridos en optimización:

1. *De refinamiento*: Este método aplica primeramente el algoritmo evolutivo. Una vez terminado el proceso evolutivo, las soluciones encontradas se mejoran mediante el método secundario elegido.
2. *De inicialización*: Este tipo de híbrido utiliza el método secundario para generar soluciones “buenas” que formarán parte de la población inicial del algoritmo evolutivo.
3. *Incrustados*: En este tipo de híbrido, el método secundario está dentro de un módulo del algoritmo evolutivo. Por ejemplo dentro de un operador de variación.

El algoritmo que proponemos cae dentro de la clase método híbrido incrustado. Ya que el método secundario está dentro del archivador propuesto. Y éste nutre a la población para seguir el proceso evolutivo.

Realizar un hibridación de estos métodos, no es un proceso trivial. Al hibridar un algoritmo evolutivo se busca por un lado acelerar el proceso de obtención

de soluciones óptimas al utilizar un algoritmo de programación matemática, y por el otro mantener los beneficios que brindan los algoritmos poblacionales, como puede ser el contar con una población de posibles soluciones. El proceso de hibridizar un algoritmo evolutivo con un método de programación matemática, que es en principio de naturaleza determinista se complica por varias razones, algunas de las que se han observado a lo largo de esta investigación son:

- *El aumento de la presión de selección.* Se origina al mejorar de manera significativa a un individuo particular dentro de la población. Esto afecta el proceso evolutivo porque los demás individuos se vuelven menos aptos para ser seleccionados para reproducirse y sobrevivir.
- *Posible pérdida de la diversidad en las soluciones.* La población tiende a converger a un único punto debido al cambio en la frecuencia de los genes en una población con el paso del tiempo y por efecto del azar, lo cual se conoce también como *desvío genético*, y como consecuencia de que algunos operadores pierden su eficacia ante una alta presión de selección.
- *Posible convergencia prematura.* Consecuencia directa del aumento de la presión de selección y de la pérdida de diversidad, usualmente la población completa se convierte en una copia del mejor individuo.
- *El aumento en el costo computacional.* Usualmente, los métodos de programación matemática son costosos computacionalmente, por ejemplo requieren un mayor número de evaluaciones de la función para obtener una buena solución. Por lo que al hibridizar, se busca que el método de programación matemática se utilice en un individuo que pueda garantizar que habrá una mejora y que no se aplique un gran número de veces, de modo que no se gaste un gran número de evaluaciones de la función. Es por esto que el método elegido para hibridizar debe ser probadamente efectivo y de bajo costo en términos de evaluaciones de la función objetivo.

## 5.2 Descripción del método

Analizando de manera teórica y empírica las diversas dificultades y ventajas de los métodos estudiados, se decidió trabajar con el método PT. En el método PT tiene como ventajas que no aumenta la no linealidad del CP, esto quiere decir que nuestro espacio solución se mantiene lineal en el proceso lo que evita el uso del corrector y además se tiene control en la distribución de las soluciones generadas, por lo cual resulta conveniente su hibridación con un método poblacional. El método poblacional elegido fue el NSGA-II. La meta principal de la hibridación de estos dos métodos son generar todo el FP con una buena distribución de soluciones con un menor costo computacional que los métodos por separado.

### 5.2.1 Módulo Archivador

Para poder introducir el método de programación matemática PT dentro del algoritmo evolutivo, se propuso un proceso denominado *Archivador* el cual se encarga de controlar tanto la aplicación del PT como la introducción de individuos mejorados dentro de la población. El archivador aquí propuesto es una adaptación de la rejilla adaptativa<sup>1</sup> del algoritmo *Pareto Archived Evolution Strategy* (PAES) [Knowles and Corne, 1999]. La función principal del archivador es revisar si cierto individuo generado es apto para ingresar al archivo y si es sujeto a refinamiento; así como de gestionar cómo se va a introducir, en caso dado, para minimizar las distorsiones dentro del proceso evolutivo. El procedimiento de archivado de soluciones propuesto se presenta en el Algoritmo 3, ahí denotamos como  $c$  al individuo candidato a entrar al archivo y como  $A$  al archivo donde se guardan las soluciones controladas. Primero se verifica que el candidato sea un elemento no dominado, posteriormente se explora la rejilla para detectar las zonas más concurridas. Este archivador genera un refinamiento de soluciones mediante el método PT además de que vigila el estado de diversidad de las soluciones.

### 5.2.2 NSGA-II + PT.

A continuación se describe el método híbrido propuesto, denotado por simplicidad como NSGA-II + PT. Se muestra el pseudocódigo general en el Algoritmo 4 y se detallan ciertos procesos internos, como son la selección y la introducción del proceso de archivado. La jerarquización de Pareto y la estimación de densidad de la población mencionadas en el Algoritmo 4, son las mismas que se emplean en el algoritmo NSGA-II original. El operador de selección del método es mediante torneo binario, pero se ha modificado para hacer restrictivo, a modo de no permitir que se seleccione el mismo padre durante la cruce. Esto evita que se reproduzca consigo mismo y disminuye el desvío genético. Este proceso de selección modificado se detalla en el Algoritmo 5. Por último el proceso de archivado usado, que emplea la malla adaptativa y el método PT es el que se describió en el Algoritmo 3. Una de las ideas originales de este trabajo es justamente la propuesta de aplicación del método de programación matemática dentro del archivador. Un aspecto notable de este diseño es la forma de preservar la diversidad dentro del archivo, en contraste con el tradicional método basado en la estimación de densidad de población. Aquí se propone utilizar una malla adaptativa que nos muestre las regiones concurridas, y de esta región se selecciona el individuo que pasará a un refinamiento mediante el buscador local.

<sup>1</sup>La rejilla adaptativa es una malla formada en el espacio objetivo que permite saber donde se localiza cada solución generada y el número de soluciones por región. La única información que se requiere para su implementación es el número de divisiones que se desean en el espacio objetivo.

<sup>2</sup>Para estimar la densidad de población se utiliza el operador de preservación de diversidad del NSGA-II que se explicó en la Sección 4.3

**Algoritmo 3** Archivador 2

---

**Entrada:**  $P_i$  : población en la generación  $i$ ;  $N = |P_i|$ ;  $A$  : archivo de soluciones;  
 $M$  : cota superior para  $|A|$ ;  $c$  : candidato a entrar en la población.

- 1: **if**  $c$  es dominado por algún miembro de  $A$  **then**
- 2:     descarta al individuo  $c$ ;
- 3: **else**
- 4:     **if**  $c$  domina a algún miembro de  $A$  **then**
- 5:         elimina de  $A$  a todos los individuos dominados por  $c$ ;
- 6:         archiva  $c$ ;
- 7:     **else**
- 8:         **if**  $|A| < M$  **then**
- 9:             archiva  $c$ ;
- 10:         **else**
- 11:             genera la malla adaptativa;
- 12:             **if**  $c$  está en una región concurrida de la malla **then**
- 13:                 aplica PT;
- 14:                 generar  $P_{aux} = A \cup$  soluciones de PT;
- 15:                 conservar las  $M$  mejores soluciones de  $P_{aux}$  seleccionadas mediante jerarquización de Pareto y estimación de densidad <sup>2</sup>.
- 16:             **else** descartar  $c$ ;
- 17:             **end if**
- 18:         **end if**
- 19:     **end if**
- 20: **end if**

---

### 5.3 Discusión

En los siguientes ejemplos se comentan y se muestran los resultados obtenidos al aplicar el método híbrido NSGA-II + PT a funciones de prueba seleccionadas. En cada caso se comparan los CP y FP obtenidos al usar el método, contra los CP y FP reales de cada POM. Los resultados obtenidos con una población de 20 individuos y un archivador  $A$  tal que  $|A| = 50$ , para cada ejemplo se trabajó con un número distinto de generaciones y evaluaciones de la función. Pero es importante notar que se reduce el costo computacional de manera notoria.

**Ejemplo 5.1.** (*Función de Dent*) Se ve en este ejemplo que el método logra generar soluciones tanto en regiones convexas como no convexas del FP. En la Figura 5.1 se puede observar el FP y CP reales con . color negro y el FP y el CP obtenidos al aplicar el método con \* color rojo. Notemos que la solución obtenida, cubre casi todo el FP real; aunque la distribución de puntos no es del todo buena. Una posible solución a esto es dejar que se ejecute el programa un mayor número de generaciones. Para la generación de este ejemplo solo se necesitaron 3,000 evaluaciones de la función para satisfacer el criterio que  $\Delta_2 < 0.01$ .

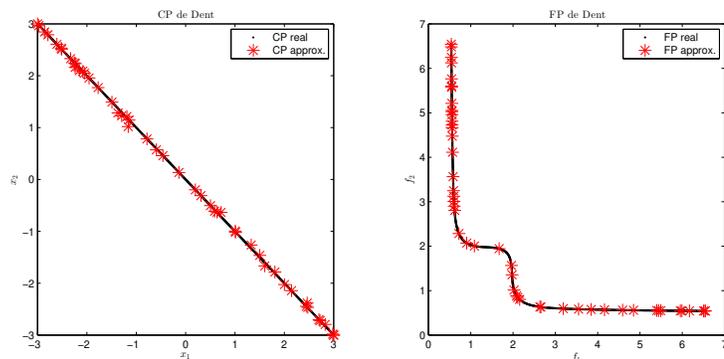
**Algoritmo 4** NSGA-II + PT

- 
- 1: Generación de la población inicial  $P_0$ .
  - 2: Jerarquización de Pareto de  $P_0$ .
  - 3: Estimación de densidad de  $P_0$ .
  - 4: **while** Criterio de Paro no se cumpla **do**
  - 5:     Selección de individuos que se reproducirán.
  - 6:     Generación de hijos a partir de cruza y mutación  $H$ .
  - 7:     Construcción de una población intermedia  $P_i = P_0 \cup H$ .
  - 8:     Jerarquización de Pareto de  $P_i$ .
  - 9:     Estimación de densidad de  $P_i$ .
  - 10:    **if** rango máx. de  $P_i = 0$  o el número de individuos no dominados es mayor al 90 % de  $P_0$ . **then**
  - 11:     Proceso de archivado  $A$ .
  - 12:    **end if**
  - 13:     Actualizar  $P_0$  a partir de  $A$ .
  - 14: **end while**
- 

**Algoritmo 5** NSGA-II + PT: Selección.

**Entrada:**  $n$  número de individuos requeridos para reproducción,  $m$  número de variables de decisión.

- 1: Generar una permutación de tamaño  $n$ , i.e, del número de individuos disponibles a seleccionar.
  - 2: **repeat** Elegir primer padre  $P_1$  correspondiente al primer elemento de la permutación.
  - 3:    **repeat** Elegir segundo padre  $P_2$  correspondiente al siguiente elemento de la permutación.
  - 4:     **until**  $P_2 \neq P_1$ .
  - 5:     Seleccionar padre que cuente con menor rango de acuerdo
  - 6:     a la jerarquización de Pareto.
  - 7:     **if** rango( $P_1$ ) = rango( $P_2$ ) **then**
  - 8:        Escoger padre con mayor valor de estimador de densidad.
  - 9:        **if** estimador( $P_1$ ) = estimador( $P_2$ ) **then**
  - 10:         Elegir  $P_1$
  - 11:        **end if**
  - 12:     **end if**
  - 13:     Añadir padre seleccionado a  $M$ .
  - 14: **until**  $M \in \mathbb{R}^{n \times m}$  este lleno.
-



**Figura 5.1:** En la primera imagen se puede observar el CP real en color negro y el CP aproximado en color rojo. En la segunda imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Ambos resultados son de la función Dent.

**Ejemplo 5.2.** (*Función ZDT1*) Recordemos que esta función muestra dificultades para la mayoría de los métodos estudiados. En la Figura 5.2 se puede observar el FP real con . color negro y el FP obtenido al aplicar el método con \* color rojo. En este ejemplo notamos que ya no hubo problema con la cola de dominancia resistente. Se cubre todo el FP real y además se cuenta con una buena distribución. Para la generación de este ejemplo solo se necesitaron 24,500 evaluaciones de la función para satisfacer el criterio que  $\Delta_2 < 0.01$ .

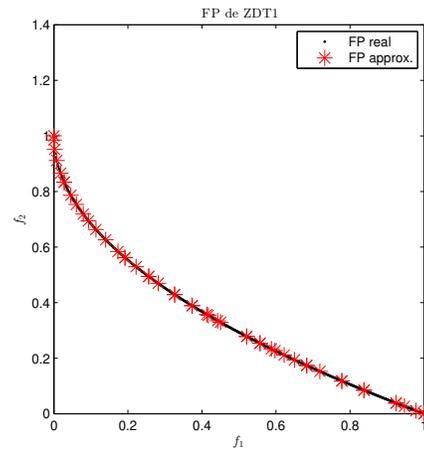
**Ejemplo 5.3.** (*Función ZDT3*) Esta función posee un FP convexo y no conectado. Se manejan 30 variables. En la Figura 5.3 se puede observar el FP real con . color negro y el FP obtenido al aplicar el método con \* color rojo. De ésta podemos observar que se consigue cubrir con una buena distribución todas las componentes del FP real. El número de componentes que cubre el método varía en ciertas ejecuciones, pero en la mayoría al menos tres de ellas se obtienen y además cuentan con una buena distribución. Para la generación de este ejemplo solo se necesitaron 23,000 evaluaciones de la función para satisfacer el criterio que  $\Delta_2 < 0.01$ .

**Ejemplo 5.4.** (*Función ZDT4*) Función de minimización que posee un FP convexo que trabaja con 10 variables. Esta función es multimodal. Se define como:

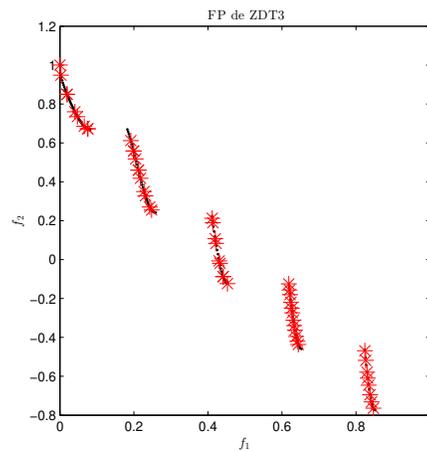
$$\text{Minimizar } F = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix}$$

con

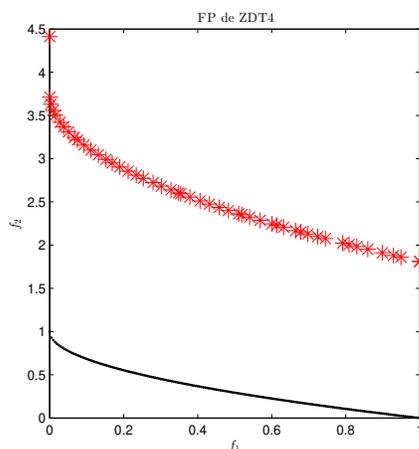
$$f_1(x) := x_1, \text{ y } f_2(x, g) := g(x) \left( 1 - \sqrt{\frac{f_1}{g(x)}} \right)$$



**Figura 5.2:** En la imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. El resultado es de la función ZDT1.



**Figura 5.3:** En la imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. Resultado correspondiente a la función ZDT3.



**Figura 5.4:** En la imagen se puede apreciar el FP real en color negro y FP aproximado en color rojo. El resultado es de la función ZDT4.

donde

$$g(x) = 1 + 10(n - 1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i))$$

con  $0 \leq x_1 \leq 1$  y  $-5 \leq x_i \leq 5$  para  $i = 2, \dots, 10$ . En la Figura 5.4 puede observar el FP real con . color negro y el FP obtenido al aplicar el método con \* color rojo. De la figura podemos observar que con esta cantidad limitada de recurso, el método genera uno de los FP locales. Aunque este no es el óptimo global, el FP que consigue cuenta con una buena distribución de soluciones. Para generar este ejemplo no se logró cumplir el criterio  $\Delta_2 < 0.01$  por el estancamiento en un FP local. El gasto computacional máximo se fijó en 60,000 evaluaciones de la función. Con estos ejemplos podemos ver un gran potencial del método propuesto. Para verificar que esto sea cierto, se realizaron 30 ejecuciones del método con algunas funciones de prueba seleccionadas. Así mismo se comparó su desempeño contra el del algoritmo NSGA-II. Estas pruebas se muestran en la siguiente sección.

Del análisis y construcción, se pueden mencionar dos ventajas prácticas observadas del método NSGA-II+PT propuesto *i) Aplicabilidad general:* dado que se logra la generación de soluciones aceptables para problemas con todo tipo de FPs, como son: convexo, cóncavo y desconectados. *ii) Eficiencia:* El algoritmo poblacional base acelera su proceso de convergencia gracias la explotación eficaz de soluciones a nivel local; haciendo computacionalmente menos costoso el proceso en términos del número de evaluaciones de la función objetivo.

En la práctica se observaron también ciertas dificultades para el algoritmo

propuesto, las cuales mencionaremos enseguida. *i) tiempo de ejecución:* debido a la comparación de puntos en el proceso de archivado, este método puede tardarse en obtener la aproximación final del FP. *ii) estancamiento heredado del PT:* La mejora en el desempeño no es del todo notable cuando se trata con funciones que cuenten con una gran cantidad de FPs locales; como se esperaba, se produce cierto estancamiento en dichos frentes debido a las características del método PT. *iii) Colas de dominancia resistente:* Se observó un proceso lento de depuración de soluciones en las colas de dominancia resistente<sup>3</sup>, esto se atribuye a algún aspecto de diseño en el método de archivado, que puede ser estudiado como trabajo a futuro. Finalmente, cabe mencionar que el método híbrido propuesto está limitado, a diferencia del algoritmo evolutivo base (NSGA-II) a que las funciones objetivo del POM a tratar sean diferenciables por partes, esto es debido a que el PT solo trabaja con funciones de este tipo. Sin embargo, esta limitación no parece ser importante en la práctica. Recordemos que nos interesa hacer un híbrido para acelerar la convergencia del método poblacional al FP real y que éste sea capaz de genera todo el FP con puntos bien distribuidos, hecho que no siempre ocurre con ninguno de los métodos. Se elige el PT al ser el que menor gasto por punto suele tener además que es posible controlar el tamaño de paso con el que se generan los puntos solución del PT, teniendo así cierto control en la distribución. Es importante mencionar que en ciertos casos, se gastan un poco más de evaluaciones de la función debido a que el punto que recibe el PT como inicial no siempre pertenece al FP real, cuando esto sucede el PT aplica el método de Newton Multiobjetivo para conseguir un punto perteneciente al FP real.

## 5.4 Análisis numérico

Se compara el valor promedio del indicador  $\Delta_2$  obtenido tras realizar 30 ejecuciones del método Híbrido y del NSGA-II por función. Para el método híbrido(NSGA-II + PT) se estableció como criterio de paro que el valor de  $\Delta_2 < \varepsilon$  o que el número de evaluaciones de la función fuera menor a una cota máxima  $f_{max}$ . Bajo este criterio, después de las 30 ejecuciones, para cada función se obtiene un valor promedio para  $\Delta_2$  y número de evaluaciones de la función. Para el NSGA-II, se establece como criterio de paro que el número de evaluaciones de la función sea menor a una cota máxima  $f_{max}^{hibrido}$ . En este caso,  $f_{max}^{hibrido}$  es el mismo número de evaluaciones de la función empleado por el método híbrido. Análogamente en cada ejecución se tiene un valor promedio para  $\Delta_2$  y número de evaluaciones de la función. De este modo el gasto computacional, en cuanto a número de evaluaciones de la función, es el mismo para ambos métodos. Se realizaron las prueba a todas las funciones del apéndice 1.

<sup>3</sup>Las colas de dominancia resistente son soluciones que en ciertos problemas sobreviven a lo largo de un gran número de generaciones aunque cualitativamente sean inferiores a otras, esto ocurre pues encontrar la solución que las domine a todas ellas es una labor muy difícil[Kita et al., 1996].

**Tabla 5.1:** Tabla de valores para los parámetros  $\varepsilon$ .

	$\varepsilon$
<b>ZDT</b>	0.02
<b>Restantes</b>	0.01

**Tabla 5.2:** Costo computacional.

<b>Función</b>	# Evaluaciones de la función promedio
<b>Binh</b>	60,000
<b>Dent</b>	2,000
<b>Fonseca</b>	15,000
<b>Laumanns</b>	15,000
<b>Lis</b>	6,000
<b>Murata</b>	2,000
<b>Poloni</b>	12,000
<b>Rendon 2</b>	5,000
<b>Schaffer 1</b>	7,000
<b>Schaffer 2</b>	7,000
<b>ZDT1</b>	42,000
<b>ZDT2</b>	21,000
<b>ZDT3</b>	60,000
<b>ZDT4</b>	60,000

Para las prueba se establecieron dos valores para  $\varepsilon$ , uno fue empleado para la familia de funciones ZDT y otro para las funciones restantes. En la Tabla 5.1 se muestran los valor establecidos. Por último  $f_{max}$  se fijó en 60,000 evaluaciones de la función.

## 5.5 Resultados

En esta sección se muestran los resultados obtenidos para el experimento. En la Tabla 5.2 se muestra el costo computacional promedio por función de prueba. Se denota como costo computacional al número de evaluaciones de la función empleadas.

En la Tabla 5.3 se muestran los resultados obtenidos para  $\Delta_2$ . Se muestra el valor promedio, el mejor y el peor valor obtenidos. Observe que para las funciones de *Dent*, *Lis*, *Murata*, *Rendon 2*, *Schaffer 1* el método híbrido obtiene

un mejor valor promedio de  $\Delta_2$  para el costo computacional determinado, es importante notar que NSGA-II funciona adecuadamente para estos ejemplos también. En el caso de *Dent* es importante notar que el híbrido logra corregir los problemas de generación de colas que el PT presenta. En cambio para las funciones *Fonseca* y *Laumanns* el método híbrido también obtiene mejor valor promedio de  $\Delta_2$ , pero el valor del indicador para el NSGA-II es lejano en comparación al del híbrido. Para la función *Binh* ambos métodos obtienen buenos resultados, pero el NSGA-II muestra una pequeña ventaja sobre el híbrido. Note que para las funciones con frentes desconectados como lo son *Poloni*, *Schaffer 2* y *ZDT3* para el número de evaluaciones utilizado el NSGA-II obtiene mejor valor  $\Delta_2$ , esto ocurre por el buscador local empleado. Recuerde que el PT genera puntos de KKT y en estos casos puede generar puntos en una componente que son dominados por soluciones pertenecientes a otras componentes del frente, cosa que no se detecta si dichos puntos no han sido generados, este hecho afecta directamente el valor del indicador. Dentro de las 30 ejecuciones hay ocasiones donde el híbrido logra generar todas las componentes con soluciones bien distribuidas a un bajo costo computacional, pero el valor promedio se ve afectado por aquellas ocasiones donde ocurre el hecho anteriormente explicado. Por otro lado para las funciones *ZDT1* y *ZDT2* el método híbrido obtiene un mejor valor del indicador, aunque es un tanto costo computacionalmente para ambos métodos. En estos casos, el método híbrido en ocasiones logra converger más rápido y con mejor distribución de soluciones debido al archivador empleado. Para la función *ZDT3*, es importante agregar que no se ve un mejor desempeño a grandes rasgos puesto que nuestro criterio de paro  $\Delta_2$  es muy restrictivo. Si se relaja este criterio, es decir  $\varepsilon$  menos pequeño, se aprecia la ventaja del método híbrido pues con pocas evaluaciones logra converger al FP real con una distribución de soluciones aceptable a diferencia del NSGA-II. Por último para la función *ZDT4*, el método híbrido consigue un mejor valor  $\Delta_2$ . En cuanto a distribución de soluciones, el método híbrido funciona muy bien. El problema que se presenta es que el estancamiento en un FP local; cosa que también ocurre con el NSGA-II. La diferencia principal en este ejemplo, es que si al NSGA-II se le permite emplear más recursos computacionales eventualmente logrará llegar al FP global, esto no ocurre con el híbrido. Si al híbrido se le permite emplear más recursos, seguirá refinando las soluciones solamente. El problema de la multifrontalidad es intrínseco del método PT. Podemos concluir que el método híbrido permite acelerar la convergencia a la solución, aunque presenta problemas con aquellas funciones que cuentan con frentes desconectados. En este tipo de funciones, el método híbrido solventa la deficiencia del PT que solo genera alguna de las componentes del frente pero no lo hace de la manera óptima. En ciertas ejecuciones el método híbrido presenta problemas en este caso pues el PT genera puntos de KKT que no forman parte del FP real lo que genera un costo computacional innecesario, para el PT este tipo de puntos son buenos pues son puntos de KKT pero no son buenos para el poblacional ya que existe un punto del FP real que lo domina.

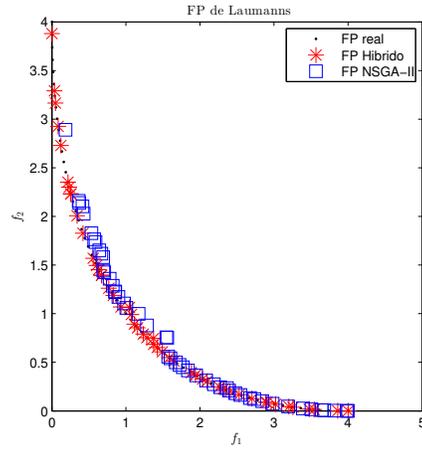
Para ejemplificar el desempeño del método híbrido contra el NSGA-II se

Tabla 5.3: Resultados para  $\Delta_2$ .

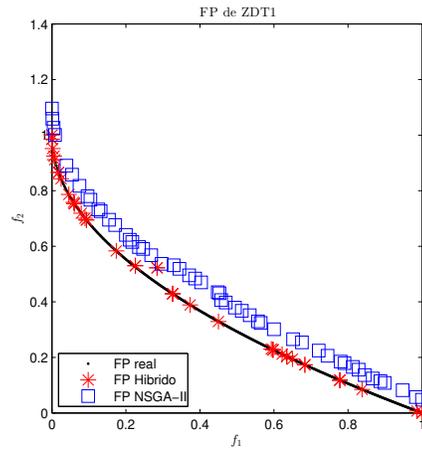
Funciones	$\Delta_2$					
	NSGA-II+PT			NSGA-II		
	Promedio	Mejor	Peor	Promedio	Mejor	Peor
<b>Binh</b>	0.0353	0.024	0.4131	<b>0.0263</b>	0.0225	0.0322
<b>Dent</b>	<b>0.0055</b>	0.0033	0.0124	0.0068	0.0026	0.0221
<b>Fonseca</b>	<b>0.0013</b>	0.0008	0.0034	0.0107	0.0006	0.0298
<b>Laumanns</b>	<b>0.0082</b>	0.0057	0.0268	0.6145	0.0064	6.9078
<b>Lis</b>	<b>0.0049</b>	0.0014	0.0094	0.0086	0.0020	0.0568
<b>Murata</b>	<b>0.0060</b>	0.0010	0.0146	0.0077	0.0010	0.0311
<b>Poloni</b>	0.0785	0.0046	0.4213	<b>0.0152</b>	0.0044	0.0656
<b>Rendon 2</b>	<b>0.0118</b>	0.0034	0.1123	0.0169	0.0035	0.1033
<b>Schaffer 1</b>	<b>0.0172</b>	0.0043	0.1070	0.0411	0.0034	0.9543
<b>Schaffer 2</b>	0.0248	0.0035	0.2231	<b>0.0034</b>	0.0017	0.0209
<b>ZDT1</b>	<b>0.0098</b>	0.0006	0.1503	0.0153	0.0060	0.0703
<b>ZDT2</b>	<b>0.0270</b>	0.0056	0.1981	0.0892	0.0218	0.2163
<b>ZDT3</b>	0.0645	0.0109	0.3933	<b>0.0126</b>	0.0025	0.0552
<b>ZDT4</b>	<b>5.5223</b>	2.8217	8.4957	8.8018	6.7610	11.2209

muestran algunas figuras. En estas figuras se muestra el FP real con  $\bullet$  de color negro, el FP aproximado por el método híbrido con  $*$  de color rojo y el FP aproximado por el NSGA-II con  $\square$  de color azul. Los resultados corresponden a las 50 mejores soluciones encontradas de las 30 ejecuciones. Se muestran las imágenes de las funciones: Laumanns, ZDT1, ZDT2 y ZDT3.

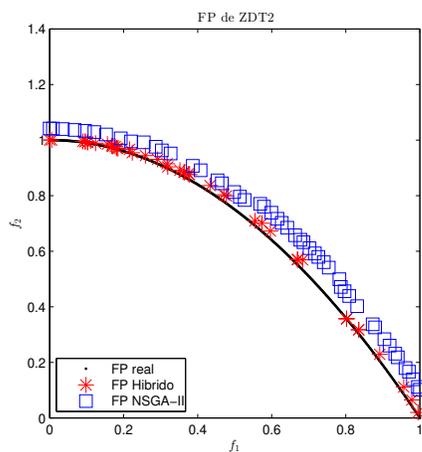
Observe que en la Figura 5.5 ambos métodos convergen al FP real, pero el método híbrido lo realiza con una mejor distribución de soluciones. Hecho que se refleja en el valor promedio de  $\Delta_2$ . Por otra parte, note que en la Figura 5.6 y en la Figura 5.7 el método híbrido obtiene mejores resultados, puesto que la aproximación está sobre la solución real mientras que el NSGA-II bajo ese esfuerzo computacional aún no lo consigue. Dado que no es muy lejana esta aproximación y los puntos obtenidos como solución cuentan con una distribución aceptable, el valor de  $\Delta_2$  es pequeña, sobre todo en ZDT1. Por último en la Figura 5.8 observe que el método híbrido cuenta con una mejor aproximación que el NSGA-II por sí solo. Aunque queda claro que el método híbrido puede mejorar, sobre todo en aquellas funciones con frentes no conectados. Algunas mejoras a realizar son la distribución de soluciones pues siguen ocurriendo aglomeraciones de soluciones y la detección oportuna de puntos que son de KKT pero que no forman parte del frente real.



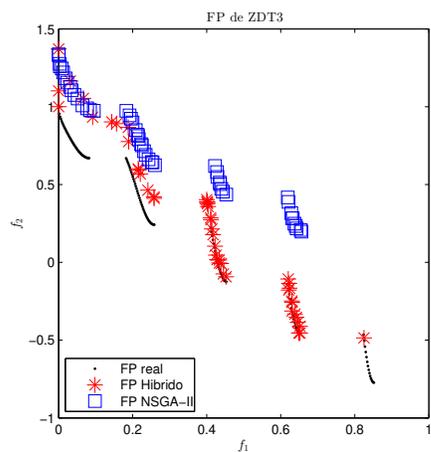
**Figura 5.5:** En la imagen se puede apreciar el FP real en . color negro, el FP aproximado por NSGA-II en □ color azul y el FP aproximado por *NSGA-II + PT* en \* color rojo. Resultados comparativos de la función Laumanns.



**Figura 5.6:** En la imagen se puede apreciar el FP real en . color negro, el FP aproximado por NSGA-II en □ color azul y el FP aproximado por *NSGA-II + PT* en \* color rojo. Resultados comparativos de la función ZDT1.



**Figura 5.7:** En la imagen se puede apreciar el FP real en  $\cdot$  color negro, el FP aproximado por NSGA-II en  $\square$  color azul y el FP aproximado por *NSGA-II + PT* en  $*$  color rojo. Resultados comparativos de la función ZDT2.



**Figura 5.8:** En la imagen se puede apreciar el FP real en  $\cdot$  color negro, el FP aproximado por NSGA-II en  $\square$  color azul y el FP aproximado por *NSGA-II + PT* en  $*$  color rojo. Resultados comparativos de la función ZDT3.

## Capítulo 6

# Conclusiones y Trabajo Futuro.

En este trabajo se presentó un análisis de cinco métodos disponibles en la literatura para resolver POMs. Los métodos estudiados se clasificaron en dos grandes grupos: *i) métodos exactos* y *ii) métodos heurísticos*. Los métodos exactos son métodos de programación matemática, con frecuencia son de naturaleza local, y utilizan información de primer y/o segundo orden. Por otro lado, los métodos heurísticos estudiados son métodos estocásticos, suelen ser de alcance global, y de propósito general. Durante el desarrollo de los capítulos previos se ha observado que cada metodología estudiada posee ciertas ventajas y desventajas, para diferentes clases de problemas. Por ejemplo, el SP funciona muy bien en funciones con geometría convexa del FP, mientras que falla en regiones no convexas del mismo. La gran ventaja de este método (el más popular en la práctica) es que garantiza que las soluciones que devuelven son óptimos de Pareto. Las desventajas de este método suelen erradicarse con el SPA, aunque se observó con los experimentos desarrollados que este último es computacionalmente costoso. Por su parte, el método NBI presenta peculiaridades. Por ejemplo en el caso del problema ZDT1, su buen desempeño está ligado a la correcta elección de los puntos que forman la CHIM, esto se deriva de la multimodalidad de alguna de las funciones objetivo. En el caso del problema ZDT3, que presenta un FP desconectado, ninguno de los métodos logra el objetivo que es generar todo el FP con una buena distribución. En este ejemplo el PT genera una buena distribución de puntos de KKT, sin embargo una desventaja es que devuelve puntos que no son parte del FP; además que solo genera una de las componentes del FP. Por otro lado, el NBI devuelve puntos en la frontera que no son parte de la solución del problema. Además que no genera una buena extensión del FP. Sin embargo, el método NBI genera buenas soluciones aunque en ocasiones es más costoso que el PT. Durante el análisis de los métodos exactos se obtuvo que en

la mayoría de los casos de prueba el método PT devolvió los mejores resultados. Este método es altamente competitivo con respecto al método NBI, en cuanto a desempeño y costo computacional. Finalmente, se concluye que si se utilizara un método exacto para resolver el POM en caso de desconocer la forma que posee su FP o en caso de que el problema no sea diferenciable sería el NBI. Y en caso de contar con al menos una solución del problema y tener garantía de la diferenciabilidad del problema, se recomienda el uso del PT.

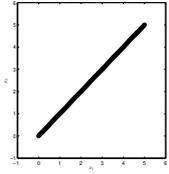
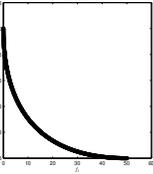
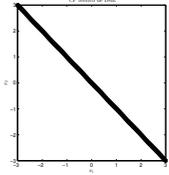
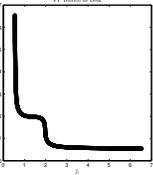
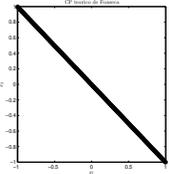
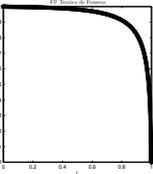
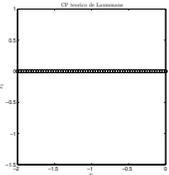
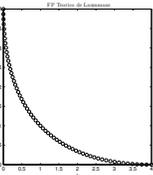
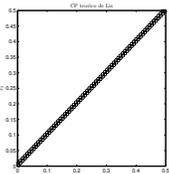
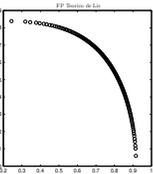
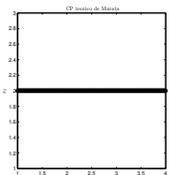
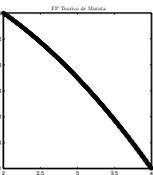
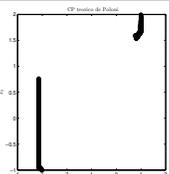
En cuanto a métodos heurísticos, se estudio el método NSGA-II. Este método es de naturaleza global, lo cual es una gran ventaja puesto que con una sola ejecución logramos generar una aproximación de la solución. Por la metodología que sigue, las soluciones que se van obteniendo buscan tener una buena distribución aunque en ocasiones obtener esto es costoso computacionalmente. Se observó que en la familia de las *ZDT* el NSGA-II tarda en converger a la solución en la mayoría de las ejecuciones. También se propuso en el capítulo 5 un método híbrido original que basado en una heurística de propósito general, *i.e.*, el NSGA-II, y un buscador local (*i.e.* el PT) logró generar el FP completo, con una buena distribución de puntos y con un bajo costo de evaluaciones de la función (comparado con el NSGA-II), para diversos problemas de prueba, acelerando así la convergencia de la heurística a la solución del POM. Si se decide utilizar métodos heurísticos, se recomienda el uso del método híbrido propuesto. Aunque por el buscador local que tiene, se necesita que el problema a solucionar sea diferenciable. En caso de que no se cumpla este supuesto, el NSGA-II representa en una buena opción. De los experimentos mostrados pudimos observar que en 70 % de los ejemplos el método híbrido obtuvo mejores resultados en el indicador de desempeño que el NSGA-II (reduciendo hasta un 40 % el costo computacional, en algunos casos). Esto se debe al método implementado para el archivado de soluciones, que utiliza el PT como buscador local. El método híbrido propuesto funciona para diversas geometrías del FPs (*e.g.*, convexo, no convexo y desconectados). Su buen desempeño se aprecia en la *ZDT3*, al lograr generar más de una componente del FP real en la mayoría de los casos.

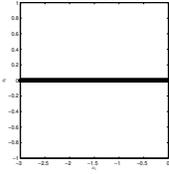
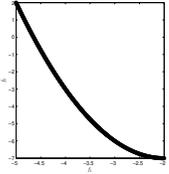
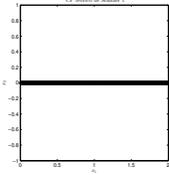
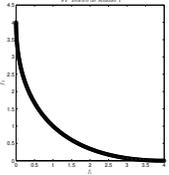
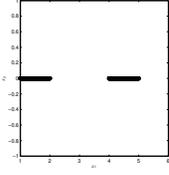
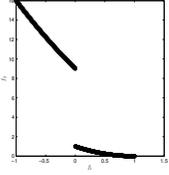
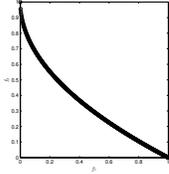
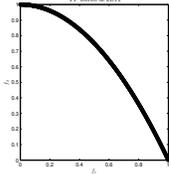
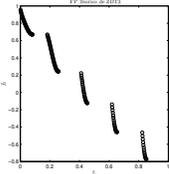
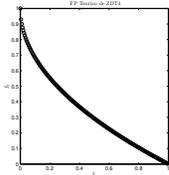
Una deficiencia del método propuesto es que aún muestra estancamiento en ciertos FPs locales, para problemas altamente multimodales (multifrontales), esto era de esperarse pues el PT utiliza información de primer y/o segundo orden de las funciones objetivo. Otra deficiencia que se detectó en el método híbrido es que no detecta de manera oportuna aquellos puntos de KKT que no pertenecen al FP real, lo que implica un aumento innecesario en el costo computacional. Ambos aspectos requieren de más trabajo en el diseño del proceso de archivado de soluciones y su interacción con el método matemático de naturaleza local, y se dejan como trabajo a futuro. Otras posibles líneas de investigación que se derivan como extensión de este trabajo son: 1) Generalizar el método de archivado propuesto para que se acople con otras heurísticas. 2) Utilizar un buscador local que nos permita trabajar con funciones no diferenciables y 3) Mejorar su funcionamiento cuando el POM cuente con colas de soluciones resistentes a la dominancia.

# Apéndice 1.

Función	Definición	Restricciones
<b>Binh</b>	$f_1(x_1, x_2) = x_1^2 + x_2^2$ $f_2(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 5)^2$	$-5 \leq x_1, x_2 \leq 10$
<b>Dent</b>	$f_1(x_1, x_2) = \frac{1}{2} \left( \sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} \right)$ $+ \frac{1}{2} (x_1 - x_2) + \lambda \exp - (x_1 - x_2)^2$ $f_2(x_1, x_2) = \frac{1}{2} \left( \sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} \right)$ $+ \frac{1}{2} (-x_1 + x_2) + \lambda \exp - (x_1 - x_2)^2$	$-3 \leq x_1, x_2 \leq 3$
<b>Fonseca</b>	$f_1(x_1, x_2) = 1 - \exp \left\{ -(x_1 - 1)^2 - (x_2 + 1)^2 \right\}$ $f_2(x_1, x_2) = 1 - \exp \left\{ -(x_1 + 1)^2 - (x_2 - 1)^2 \right\}$	Ninguna
<b>Laumanns</b>	$f_1(x_1, x_2) = x_1^2 + x_2^2$ $f_2(x_1, x_2) = (x_1 + 2)^2 + x_2$	$-50 \leq x_1, x_2 \leq 50$
<b>Lis</b>	$f_1(x_1, x_2) = \sqrt[8]{x_1^2 + x_2^2}$ $f_2(x_1, x_2) = \sqrt[4]{(x_1 - 0.5)^2 + (x_2 - 0.5)^2}$	$-5 \leq x_1, x_2 \leq 10$
<b>Murata</b>	$f_1(x_1, x_2) = 2\sqrt{x_1}$ $f_2(x_1, x_2) = x_1(1 - x_2) + 5$	$1 \leq x_1 \leq 4$ $1 \leq x_2 \leq 2$
<b>Poloni</b>	$f_1(x_1, x_2) = -[1 + (A_1 - B_1)^2 + (A_2 - B_2)^2]$ $f_2(x_1, x_2) = -[(x_1 + 3)^2 + (x_2 + 1)^2]$ $A_1 = 0.5 \sin 1 - 2 \cos 1 + \sin 2 - 1.5 \cos 2,$ $A_2 = 1.5 \sin 1 - \cos 1 + 2 \sin 2 - 0.5 \cos 2,$ $B_1 = 0.5 \sin x - 2 \cos x + \sin y - 1.5 \cos y,$ $B_2 = 1.5 \sin x - \cos x + 2 \sin y - 0.5 \cos y.$	$-\pi \leq x_1, x_2 \leq \pi$

Función	Definición	Restricciones
<b>Rendón 2</b>	$f_1(x_1, x_2) = x_1 + x_2 + 1$ $f_2(x_1, x_2) = x_1^2 + 2x_2 - 1$	$-3 \leq x_1, x_2 \leq 3$
<b>Schaffer 1</b>	$f_1(x) = x^2$ $f_2(x) = (x - 2)^2$	Ninguna
<b>Schaffer 2</b>	$f_1(x) = \begin{cases} -x & \text{si } x \leq 0 \\ -2 + x & \text{si } 1 < x \leq 3 \\ 4 - x & \text{si } 3 < x \leq 4 \\ -4 + x & \text{si } x > 4 \end{cases}$ $f_2(x) = (x - 5)^2$	$-5 \leq x \leq 10$
<b>ZDT1</b>	$f_1(x) = x_1$ $f_2(x, g) = g(x) * \left(1 - \sqrt{\frac{1}{g(x)}}\right)$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^{30} x_i$	$0 \leq x_i \leq 1$ $i = 1, \dots, m$
<b>ZDT2</b>	$f_1(x) = x_1$ $f_2(x, g) = g(x) * \left(1 - \left(\frac{f_1}{g(x)}\right)^2\right)$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^{30} x_i$	$0 \leq x_i \leq 1$ $i = 1, \dots, m$
<b>ZDT3</b>	$f_1(x) = x_1$ $f_2(x, g) = g(x) \left(1 - \sqrt{\frac{f_1}{g(x)}} - \frac{f_1}{g(x)} \sin(10\pi f_1)\right)$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^{30} x_i$	$0 \leq x_i \leq 1$ $i = 1, \dots, m$
<b>ZDT4</b>	$f_1(x) = x_1$ $f_2(x, g) = g(x) \left(1 - \sqrt{\frac{f_1}{g(x)}}\right)$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i))$	$0 \leq x_1 \leq 1$ $-5 \leq x_i \leq 5$ $i = 2, \dots, m$

<p><b>Binh</b></p>		
<p><b>Dent</b></p>		
<p><b>Fonseca</b></p>		
<p><b>Laumanns</b></p>		
<p><b>Lis</b></p>		
<p><b>Murata</b></p>		
<p><b>Poloni</b></p>		

<p><b>Rendon 2</b></p>		
<p><b>Schaffer 1</b></p>		
<p><b>Schaffer 2</b></p>		
<p><b>ZDT1</b></p>		
<p><b>ZDT2</b></p>		
<p><b>ZDT3</b></p>		
<p><b>ZDT4</b></p>		

# Bibliografía

- [Bäck, 1996] Bäck, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.
- [Batista et al., 2009] Batista, L. S., Guimarães, F. G., Ramírez, J., et al. (2009). A differential mutation operator for the archive population of multi-objective evolutionary algorithms. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 1108–1115. IEEE.
- [Belegundu et al., 1994] Belegundu, A., Murthy, D., Salagame, R., and Constant, E. (1994). Multi-objective optimization of laminated ceramic composites using genetic algorithms. In *A Collection of Papers, 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, volume 2, pages 1015–1022.
- [Bhaskar et al., 2000] Bhaskar, V., Gupta, S. K., and Ray, A. K. (2000). Applications of multiobjective optimization in chemical engineering. *Reviews in chemical engineering*, 16(1):1–54.
- [Coello and Cortés, 2005] Coello, C. A. C. and Cortés, N. C. (2005). Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, 6(2):163–190.
- [Coello and Zacatenco, 2004] Coello, C. A. C. and Zacatenco, C. S. P. (2004). Introducción a la computación evolutiva (notas de curso). *CINVESTAV-IPN, Departamento de Ingeniería Eléctrica, Sección de Computación. México, DF*.
- [Darwin, 1929] Darwin, C. (1929). *The Origin of Species by Means of Natural Selection or The Preservation of Favored Races in the Struggle for Life*. The Book League of America.
- [Das and Dennis, 1997] Das, I. and Dennis, J. E. (1997). A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural optimization*, 14(1):63–69.

- [Das and Dennis, 1998] Das, I. and Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multi-criteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657.
- [Deb et al., 2000] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. *Lecture notes in computer science*, 1917:849–858.
- [Ehrgott, 2005] Ehrgott, M. (2005). *Multicriteria optimization*, volume 491. Springer Verlag.
- [Eiben, 1998] Eiben, A.E & Smith, J. (1998). *Introduction to Evolutionary Computing*. Springer.
- [Figueira et al., 2005] Figueira, J., Greco, S., and Ehrgott, M. (2005). *Multiple criteria decision analysis: state of the art surveys*, volume 78. Springer Verlag.
- [Fliege et al., 2009] Fliege, J., Drummond, L. G., and Svaiter, B. F. (2009). Newton’s method for multiobjective optimization. *SIAM Journal on Optimization*, 20(2):602–626.
- [Fogel, 1966] Fogel, L. (1966). *Artificial Intelligence through Simulated Evolution*. John Wiley & Sons, Inc.
- [Fogel, 1965] Fogel, J. O. A. . W. M. (1965). Artificial intelligence through a simulation of evolution. In *In M. Maxfield, A. Callahan, and L. J. Fogel, editors, Bio-physics and Cybernetic Systems: Proceedings of the Second Cybernetic Sciences Symposium.*, pages 131–155. Spartan Books.
- [Hájek et al., 2010] Hájek, J., Szöllös, A., and Šístek, J. (2010). A new mechanism for maintaining diversity of pareto archive in multi-objective optimization. *Advances in Engineering Software*, 41(7):1031–1057.
- [Hillermeier, 2001] Hillermeier, C. (2001). *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*, volume 135 of *International Series of Numerical Mathematics*. Birkhäuser.
- [Holland, 1962a] Holland, J. (1962a). Concerning efficient adaptive systems. In *In M. C. Yovits, G. T. Jacobi, and G. D. Goldstein.*, pages 215–230. Spartan Books.
- [Holland, 1962b] Holland, J. (1962b). Outline for a logical theory of adaptive systems. *Journal of the Association for Computing Machinery.*, 9:297–314.
- [Hu and Yen, 2013] Hu, W. and Yen, G. G. (2013). Density estimation for selecting leaders and mantaining archive in mopso. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 181–188. IEEE.
- [Ishibuchi and Yamamoto, 2004] Ishibuchi, H. and Yamamoto, T. (2004). Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy sets and systems*, 141(1):59–88.

- [Karush, 1939] Karush, W. E. (1939). *Minima of functions of several variables with inequalities as side conditions*. PhD thesis, University of Chicago.
- [Kim and De Weck, 2005] Kim, I. Y. and De Weck, O. (2005). Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and multidisciplinary optimization*, 29(2):149–158.
- [Kita et al., 1996] Kita, H., Yabumoto, Y., Mori, N., and Nishikawa, Y. (1996). Multi-objective optimization by means of the thermodynamical genetic algorithm. In *Parallel problem solving from nature—ppsn iv*, pages 504–512. Springer.
- [Knowles and Corne, 1999] Knowles, J. and Corne, D. (1999). The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 1. IEEE.
- [Knowles and Corne, 2000] Knowles, J. D. and Corne, D. W. (2000). M-paes: A memetic algorithm for multiobjective optimization. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 325–332. IEEE.
- [Kuhn and Tucker, 1951] Kuhn, H. and Tucker, A. (1951). Nonlinear programming. In Neumann, J., editor, *Proceeding of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492.
- [Li et al., 2012] Li, W.-x., Zhou, Q., Zhu, Y., and Pan, F. (2012). An improved mopso with a crowding distance based external archive maintenance strategy. In *Advances in Swarm Intelligence*, pages 74–82. Springer.
- [Li et al., 2014] Li, Y., Cai, X., Fan, Z., and Zhang, Q. (2014). An external archive guided multiobjective evolutionary approach based on decomposition for continuous optimization. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 1124–1130. IEEE.
- [Martinez and Coello, 2010] Martinez, S. Z. and Coello, C. A. C. (2010). An archive strategy based on the convex hull of individual minima for moeas. In *2010 IEEE Congress on Evolutionary Computation (CEC'2010)*, pages 912–919.
- [Martín, 2014] Martín, A. (2014). Pareto tracer: Método predictor corrector para problemas de optimización multiobjetivo. Master’s thesis, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional.
- [Mendel, 1865] Mendel, G. (1865). Versuche über pflanzenhybriden. abhandlungen des naturforschenden vereines in brünn 4, 3-47. traducción inglesa (1901). *Journal of the Royal Horticultural Society*, 26:1–32.
- [Michalewicz and Fogel, 2004] Michalewicz, Z. and Fogel, D. B. (2004). *How to solve it: modern heuristics*. Springer Science & Business Media.

- [Miettinen, 1999] Miettinen, K. (1999). *Nonlinear multiobjective optimization*, volume 12. Springer.
- [Motta et al., 2012] Motta, R. d. S., Afonso, S. M., and Lyra, P. R. (2012). A modified nbi and nc method for the solution of n-multiobjective optimization problems. *Structural and Multidisciplinary Optimization*, 46(2):239–259.
- [Murata et al., 1996] Murata, T., Ishibuchi, H., and Tanaka, H. (1996). Multi-objective genetic algorithm and its applications to flowshop scheduling. *Computers & Industrial Engineering*, 30(4):957–968.
- [Neri and Cotta, 2012] Neri, F. and Cotta, C. (2012). Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, 2:1–14.
- [Pareto, 1927] Pareto, V. (1971 (original edition in French in 1927)). *Manual of Political Economy*. The MacMillan Press.
- [Praditwong and Yao, 2006] Praditwong, K. and Yao, X. (2006). A new multi-objective evolutionary optimisation algorithm: the two-archive algorithm. In *Computational Intelligence and Security, 2006 International Conference on*, volume 1, pages 286–291. IEEE.
- [Quintero and Coello, 2006] Quintero, L. V. S. and Coello, C. A. C. (2006). Una introducción a la computación evolutiva y alguna de sus aplicaciones en economía y finanzas. *Revista de Métodos Cuantitativos para la Economía y la Empresa*, (2):3–26.
- [Rechenberg, 1973] Rechenberg, I. (1973). *Evolutionsstrategien: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog.
- [Rothlauf, 2011] Rothlauf, F. (2011). *Design of modern heuristics: principles and application*. Springer Science & Business Media.
- [Schütze et al., 2012] Schütze, O., Esquivel, X., Lara, A., and Coello, C. A. C. (2012). Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization. *Evolutionary Computation, IEEE Transactions on*, 16(4):504–522.
- [Sinha and Goldberg, 2003] Sinha, A. and Goldberg, D. E. (2003). A survey of hybrid genetic and evolutionary algorithms. *IlliGAL report*, 2003004.
- [Toscano Pulido, 2005] Toscano Pulido, G. (2005). On the use of self-adaptation and elitism for multiobjective particle swarm optimization.
- [Van Veldhuizen and Lamont, 1998] Van Veldhuizen, D. A. and Lamont, G. B. (1998). Evolutionary computation and convergence to a pareto front. In *Late breaking papers at the genetic programming 1998 conference*, pages 221–228. Citeseer.

- [Wang et al., ] Wang, H., Jiao, L., and Yao, X. An improved two-archive algorithm for many-objective optimization.
- [Wang, 1991] Wang, S. (1991). Second-order necessary and sufficient conditions in multiobjective programming. *Numerical Functional Analysis and Optimization*, 12(1-2):237–252.
- [Xie et al., 2008] Xie, J., Zheng, J., Luo, B., and Li, M. (2008). Applying a new grid-based elitist-reserving strategy to emo archive algorithms. In *Frontiers in Algorithmics*, pages 276–287. Springer.
- [Zitzler et al., 2000] Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195.